Handoffs Management in Follow-the-Sun Software Projects: A Case Study

Josiane Kroll
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) / Brazil
josi.unc@gmail.com

Ita Richardson
Lero - the Irish Software Engineering Research Centre/ Ireland
ita.richardson@lero.ie

Jorge L. N. Audy
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) / Brazil
audy@pucrs.br

Jude Fernandez
Infosys Technologies Ltd
Bangalore, India
JUDEF@infosys.com

Abstract

In Follow-the-sun (FTS) software projects, handoffs are performed at the beginning and the end of each working day shift. Handoff efficiency is determined by the quality of knowledge transfer and the length of time taken to handover. If handoffs are not properly managed, FTS development may not result in benefits for the project. Thus, in this study, we investigate how handoffs management should be performed in an FTS software development context. We present an experience report describing the development and management handoffs in a FTS software project. Our results describe the participants' perception about software engineering activities performed, challenges faced and solutions performed to minimize these challenges. Based on our results, we also highlight management elements for handoffs which should be useful for Project Managers.

1. Introduction

Many companies run Global Software Development (GSD) projects to benefit from cheaper, faster and better development of software systems, products and services [1]. Nowadays, companies also want to take advantage of time, expertise and talent pools, wherever they may be located in the world, as these contribute to their competitive advantage and connection to local markets [2]. Companies re-structure their IT area by extending operations to offshore software development centers. Thus, follow-the-sun (FTS) development is seen as a potential strategy for these companies.

FTS enables continuous development during working day shifts spread across the globe [2]. Its main goal is to reduce software development life cycle duration or time-to-market [3].

During FTS development, each team member works normal working hours pertaining to his/her time zone. At the end of each working day shift, the work-in-progress follows to another team member located in a different time zone. When a team finishes its working day shift in one site, another team localized in another site takes the work and starts another working day shift [3]. For each 24 hour period, there are two or more shifts [4].

However, while the FTS concept looks promising in theory, it appears to be difficult in practice [5]. According to Carmel et al. [6], FTS is difficult and uncommon because production teams are sequentially handing off work-in-progress (unfinished objects) from site to site. This transfer of work-in-progress between sites is called handoff.

Handoffs require a large coordination, communication and collaboration effort. Furthermore, handoffs must be fast and efficient in order to reduce the development cycle duration [3]. FTS efficiency is determined by both the quality of knowledge transfer and its duration [6]. Handoff coordination difficulties can negatively affect team productivity and consequently will not result in benefits for the project [7]. In addition, misunderstandings may also lead to re-work [6].

In this study, we present an experience report describing the development and management of handoffs for a FTS software project. Following FTS rules described by Carmel and Espinosa [3], handoffs were performed at the beginning and at the end of each working day shift at each production site. We collected data about these handoffs, which are presented and discussed in this paper.

The main contribution of this study is the identification of challenges faced by FTS teams when conducting handoffs and the solution applied to minimize them. In our conclusions, we also highlight management elements to support the efficiency of handoffs.

The remainder of the paper is structured as follows. Section 2 presents the main concepts discussed in this study. Section 3 presents the related work. Section 4 describes the research method utilized in this study. In section 5, we present the

978-1-4799-2504-9/14 $31.00 © 2014 IEEE
DOI 10.1109/HICSS.2014.49

obtained results. In section 6, we discuss these results. Finally, section 7 provides the conclusions.

2. Follow-the-Sun Software Development

Follow-the-sun (FTS) is a subset of Global Software Development (GSD) [8]. It is applied in the context of GSD in order to take advantage of the temporal distance between several production sites located in different time zones [9]. FTS is uniquely focused on speed of development. Its main purpose is the reduction of the software development life cycle duration or time-to-market [3].

FTS does not offer other advantages besides decreasing the duration. It is applied to software projects when a software product needs to be developed quickly and the cost is irrelevant to the client [6].

In FTS software development, as team members are distributed across multiple time zones, organizations can develop software twenty-four hours continuously. Thus, the time reduction may be by 50% if there are two sites and by 67% if there are three sites [8]. Additionally, according to Solingen and Valkema [9], when the number of sites in a daily cycle increases, on average, the overall working speed of the sites also increases.

2.1. Handoffs

At the beginning and at the end of each working day shift there is a handoff. *Handoff* is a term utilized in the literature to define the process transition from a site to another [3].

Handoffs are performed on a daily basis to present a status update and to pass on unfinished tasks (project source) from one site to another. The next site will take these tasks in order to start its working day shift [5].

Performing handoffs creates dependencies between production sites [6]. The team that will be starting the working day shift depends on the status update and project source from the last production site. In the literature, handoffs' management is mentioned as one of the main challenges implementing FTS projects [3] [10] [11].

3. Related Work

In the FTS literature, few studies have explored the handoffs' management in FTS software projects. The main study that discusses handoffs was performed by Hess and Audy [12]. In this study, the authors propose a software process for daily handoffs. This process aims to alleviate difficulties faced by teams during the development phase of FTS software projects. A controlled experiment was performed to evaluate its efficiency. Findings from this study show that is possible to reduce development difficulties in FTS using the proposed process. The process is based on Composite Persona (CP) [4] and 24hr Design and Development [13] concepts. In addition, it uses the test-driven development (TDD) technique and stand-up meeting guidelines from Scrum methodology to conduct meetings.

Other studies, such as, Carmel [14] and Ramesh and Dennis [15] report characteristics and challenges faced by teams to perform handoffs on FTS projects. These studies observe problems mainly related to daily handoffs' management.

In the study performed by Carmel [14], it was observed that many problems related to handoffs' management were due to a lack of time overlap between teams and the lack of communication in real time.

Ramesh and Dennis [15] reports challenges faced by teams managing versions of code and documents. Furthermore, it was observed in this study that many problems occurred during synchronous communication performed by telephone and email.

4. Research Methodology and Study Settings

The research question that guides this research is: *How handoffs management should be performed in a FTS software development context?* We utilized the study case research method to answer this question.

The case study was performed at Infosys Technologies in Bangalore, India. Infosys has production sites and clients localized over 30 countries. The organization is very experienced in working on distributed software projects.

The study case consisted of developing a software application in the FTS mode. We chose to study the FTS development phase, because findings from studies conducted in the software industry show that FTS is effective for the testing and development phases [6]. The application development was estimated at an effort of 832 hours (one month).

Also, following the recommendations from the literature, the company adopted agile practices from Scrum method to manage the project. According to Kroll at al. [16] and Yap [17], agile methods are the most promising way to develop FTS projects.

4.1. Team setup

The software project was developed by team members distributed in three different production sites: Mexico, India and Australia.

Australia and India have some overlapping working hours. Mexico was included to extend the working hours between sites.

Two developers from each site were allocated to the project. Also, one Project Manager and one scrum master, both from India, were allocated to the project.

The team members allocated had different experience levels. In Mexico, there were two developers trainee. In India, one Project Manager had approximately 10 years software engineering experience, but without previous experience as a Project Manager. Two Indian developers had two years experience. In Australia, there were two developers. One had eight years experience, the other fourteen years.

4.2. Handoffs' planning

To perform daily handoffs between sites, developers utilized telephone calls. They also utilized the Microsoft Office Communicator tool for communication support and a data repository for sharing code and documents. Additionally the Microsoft Office Communicator tool was provided as a resource to solve specific issues or to support handoffs in case there were connection problems between sites.

The data repository used in the project was already used by Infosys. In this data repository, all files and code from the project were shared between sites.

To support handoffs, the Project Manager created a handover template using Microsoft Excel. This template was used to save handoffs' data to help team members to transfer tasks details and project updates from one site to another. The handover template was used by team members to discuss finished tasks, tasks in progress and how tasks must be continued during the next shift.

Tasks from the project were provided in the project data repository in a backlog file. The backlog file was created by the Project Manager based on recommendations from Scrum.

Task allocation was based on the CPro concept proposed by Denny et. [4]. CPro is an agile software process that improves the CP (Composite Persona) performance. It also assigns workloads to different members of a CP in a way that maximizes productivity.

Based on CPro concept, the researchers created two CPs. Each CP comprised at least one team member from each site, as shown in Figure 1. Tasks were allocated to a CP rather than to an individual team member. Each CP should manage their own tasks allocated and tasks execution.

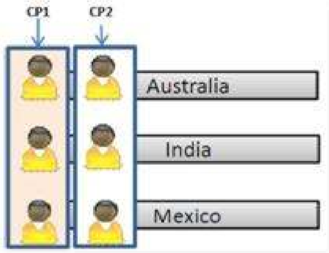


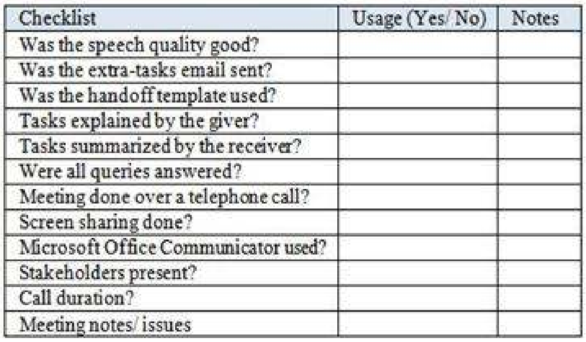
Figure 1. Composite Persona (CP) formation.

4.3. Data collection

The researchers held training meetings to clarify FTS theory among team members. Remaining questions after training meetings were clarified by email. Training meetings were also used to clarify Scrum practices, because only developers from Australia had previous experience using agile methods.

The project was estimated to be developed over a period of four weeks. Following Scrum recommendations, the project development was divided into two-week sprints. Since no standard existed for FTS project planning, the researchers used software practices already adopted by the company and Project Manager experience to plan the project. The effort hours required to develop each sprint were based on typical two-location mode development.

To manage handoffs and to ensure that FTS development rules would be followed by team members, researchers created a document with 12 checklists, as shown in Figure 2. This document was used to collect data from handoffs. This checklist was filled by researchers at the end of each handoff meeting.



Checklist	Usage (Yes/ No)	Notes
Was the speech quality good?		
Was the extra-tasks email sent?		
Was the handoff template used?		
Tasks explained by the giver?		
Tasks summarized by the receiver?		
Were all queries answered?		
Meeting done over a telephone call?		
Screen sharing done?		
Microsoft Office Communicator used?		
Stakeholders present?		
Call duration?		
Meeting notes/ issues		

Figure 2. Checklist document.

We held meetings with the Project Manager in order to identify improvements in the approach used to develop the project. The Project Manager attended handoffs between Australia and India and between India and Mexico. His role was to discuss allocated tasks, clarify project issues and manage the project progress.

At the end of sprint 1, following Scrum recommendations, the Project Manager performed the sprint 1 retrospective. The sprint 1 retrospective consisted of applying questionnaires to evaluate project activities. Its goal was to identify what was working well and what was not working in the FTS project activities. The evaluated activities were:

- Software engineering practices, standards and templates
- Handover template and process
- Communication flow (call, email, chat)
- Task allocation
- CP experience (peer interaction, work sharing, responsibility, visibility, time zone management)
- Tools used/ not used
- Portal used, openness to change, corporative tools (TFS and DeW)

In this paper, we report results related to activities of handoffs' development and management. Results obtained in other activities are reported in Kroll at al. [18].

5. Results

In this section, we report results obtained, including strategies adopted to improve handoffs development and management. We also describe participants' perceptions about software engineering activities to manage handoffs which are performed during the project.

5.1. Data from checklist document

From results obtained in the checklist document, we observed low quality communication over telephone calls due to different accents and languages. To improve the communication quality between team members, we identified two changes in the communication protocol:

- Team members must speak slowly and clearly in order to reduce accents and improve the communication;
- Team members must use appropriate language for the context, avoiding slang or unknown terms for the area.

With relation to extra tasks, the Project Manager allocated these by email. We identified some problems such as misunderstanding of extra tasks. Two strategies to minimize this problem were adopted by the team:

- Extra tasks allocated by Project Manager should be discussed during handoffs before weekends and holidays;
- As an additional resource, clarifications about extra tasks should be sent by email. The handover template contributed to manage handoffs' information. The team did not report any problem in using this template.

We also did not identify problems related to the use of the CPro concept. By following the handover template, each CP pair discussed tasks and the next steps. To clarify code and documents with another site, team members used screen sharing.

In some handoffs, we observed that a CP member was not present. In these cases, the CP member (giver) sent an email with handoff information to the next CP. Thus, we observed that to address the unavailability of a CP member, an additional email with all handoff information could be sent to the next site.

We also observed handoffs duration between sites. In Table 1, we present data from handoffs duration in sprint 1. These data were collected automatically by Infosys using a telephone call center system.

Table 1. Handoffs time duration (Sprint 1).

Date	Location	Duration (minutes)
7-Aug	India – Mexico	38
7-Aug	Australia – India	49
7-Aug	Mexico – Australia	19
6-Aug	India – Mexico	34
6-Aug	Mexico – Australia	65
3-Aug	India – Mexico	44
3-Aug	Australia – India	34
3-Aug	Mexico – Australia	16
2-Aug	India – Mexico	43
2-Aug	Australia – India	44
2-Aug	Mexico – Australia	16
1-Aug	India – Mexico	37
1-Aug	Australia – India	50
1-Aug	Mexico – Australia	39
31-Jul	India – Mexico	55
31-Jul	Australia – India	69
31-Jul	Mexico – Australia	24
30-Jul	India – Mexico	53
30-Jul	Australia – India	45
27-Jul	India – Mexico	48

27-Jul	Australia – India	38
27-Jul	Mexico – Australia	18
26-Jul	India – Mexico	37
26-Jul	Australia – India	20
26-Jul	Mexico – Australia	11
Average between sites	India – Mexico	39.78
	Australia – India	39
	Mexico – Australia	37.34
Average (all meetings)	37.84	
Total minutes	946	
Total hours	15hrs76 min	

Handoffs among sites were performed for 38 minutes on average in sprint 1. The longest handoff was performed by team members from Australia and India. It occurred because team members from these locations used handoffs to plan the next tasks. The longest handoff took 1 hour and 10 minutes. The shortest handoff was between Mexico and Australia. It took 11 minutes.

In literature, studies recommend handoffs duration of 30 minutes [12]. However, our findings show some handoffs were performed for more than 30 minutes. Thus, we observe some strategies to reduce handoffs duration:

- Distributing the communication time between CPs
- Creating rules for communication during handoffs

All handoffs were performed by telephone enhanced at times by the use of Microsoft Office Communicator. We observed that the main benefit of Microsoft Office Communicator was the screen sharing resource. Thus, we recommend screen sharing as an additional communication resource.

5.2. Project Manager meetings

The FTS project was a new experience for all of the FTS team and especially for the Project Manager. The Project Manager did not have previous experience in this role. However, his experience working on distributed software projects contributed to add improvements to the activities performed during the project.

The researches held four meetings in which the Project Manager was interviewed about the process and software practices included in the process. These meetings were performed during sprint 1. Sprint 1 was estimated to be developed over two weeks duration. Thus, researchers performed meetings at the beginning and at the end of each week.

The main contributions identified from these meetings are described below:

- *CP owner attribution:* at the beginning of the project, some tasks allocated to CPs were started, but were not finished. This was due to no one person being responsible for the tasks. Thus, to ensure that tasks will be started and finished by CPs, the Project Manager assigned, by email, a CP owner for each task. When a task was completed, each CP should inform the Project Manager. This also helped to manage task allocation.
- *Task allocation per working day:* following Scrum practices, the Project Manager used the sprint backlog file to describe the tasks and its allocation to each CP. However, team members reported difficulties in understanding the sprint backlog file. Therefore, the Project Manager created a daily email to inform CPs about the task allocated. The task allocation per working day contributed to the definition of priorities and reduction of problems faced by teams when categorizing a task in the sprint backlog file.
- *Extra tasks:* the Project Manager took advantage of information provided by CP owners related to tasks progress to verify the necessity of new task assignments for the working day. When necessary, the Project Manager assigned extra tasks by email.

5.3. Sprint 1 retrospective

Findings obtained in the sprint 1 retrospective show positive evaluations for the use of the handover template. This same result was found in the checklist document.

The Communication flow, which include tools used to perform synchronous and asynchronous communication during the project was also positively evaluated. However, team members reported difficulties related to English language skills and different experience levels.

The data repository used to share documents and code contributed to the controlling code and document versions. In addition, the data repository encouraged the punctuality of team members regarding file upload. Participants also mentioned the lack of experience related to the use of the portal at the beginning of the project.

Researchers also observed some advantages and disadvantages indicated by the FTS team. Advantages are related to team punctuality, intensive communication and good time management. Disadvantages identified are the lack of standards and templates to develop activities, inappropriate task allocation at the beginning of the project and lack of project guidelines.

6. Discussion

Relatively few studies discussing handoffs development and management have been published. The lack of studies discussing handoffs and their characteristics could be a barrier for companies interested in implementing projects using FTS. We observed many challenges faced by the FTS team for conducting this study as they had little information available to provide practical suggestions as to how to perform handoffs.

6.1. Challenges

In the FTS literature many challenges are reported which relate to coordination, communication and culture [6] [11]. We observed that these challenges can increase or decrease according to the project settings.

In our study, the FTS project settings were far from ideal because they have constrains. These constraints are related to the company size, the background and skills of its staff, the customer and market requirements and the company culture [19]. However, it is important to mention that all software development environments have constraints. In addition, these constraints are similar in many software projects.

We observed that these constraints can result in challenges that substantially affect on handoffs efficiency in FTS projects. The main challenges identified for handoffs management in this study are lack of language skills, task allocation issues, handoffs during weekends and holidays, different work experience levels and lack of trust among team members.

1. Lack of language skills

We observed that lack of language skills can become a barrier to sharing knowledge during handoffs. According to Richardson et al. [20], English is the business language required during GSD projects. However, many co-workers will not have this as their first language and misunderstandings can occur.

2. Task allocation issues

In relation to task allocation, we observed the lack of clear criteria for task allocation in FTS projects. It was followed the CPro concept for task allocation and practices from the Scrum method. However, we identified difficulties in managing the progress on tasks and in allocating extra tasks.

3. Handoffs during weekends and holidays

At the beginning of the project, researchers created a template in order to save and transfer handoff information. This template worked very well during sequential handoffs. However, after weekends and during holidays when there is no sequential handoff, the information provided in the template was not enough for working continuity. According to Carmel at al. [6], handoff efficiency depends on the quality of knowledge transfer. We identified handoffs during weekends and holidays as a big challenge for handoffs management.

Deshpande and Richardson [21] recommend backup teams as a possible solution for handoffs during weekends and holidays. However, our experience in this study showed that backup teams could be inappropriate for some FTS projects. This is because of legislation restrictions preventing people from some countries exceeding a certain number of hours per working day or week. Moreover, the number of extra work hours can dramatically raise the project budget making it unviable [22].

4. Different working experience levels

Human resources allocated to develop a FTS project may have different working experience levels. An imbalanced experience level affects the team productivity. According to Denny et al. [23] differences in skills and experience can create large variation in productivity. More experienced team members are able to develop tasks and solve problems faster than less experienced team members. Moreover, team members have different skills and means to develop the same task.

During handoffs, the imbalance experience level of the team members resulted in misunderstandings of the information transmitted, increase in handoffs duration and loss of trust between team members. We observed that team members need to develop new abilities to address new requirements or innovations within FTS environments.

5. Lack of trust among team members

Many factors can affect the trust among team members in FTS projects. According to Denny et al. [4], trust is notoriously difficult to establish among teams. Moreover, humans find hard to trust those that they have not physically met [17].

Our results show that cultural differences and different experience levels as strong factors for decreasing trust among team members.

6.2. Management elements for handoffs

A number of management elements can be drawn from our results in this study. These elements must be considered by managers in order to improve handoffs' efficiency.

1. Improving communication quality

There are many tools and software available which allow synchronous and asynchronous communication between distributed teams to be performed [15]. The team must be familiar with these resources in order to improve communication quality. Moreover, language issues must be solved before a project starts.

2. Ensure knowledge transfer between team members and production sites

During handoffs, team members exchange information about finished and in progress tasks and project updates. We observed that summarizing this information by teams' receivers at the end of handoffs contributes to minimizing isunderstandings.

3. Management of handoff's information

The handover template was utilized to save information and support teams during handoffs. The solution in this case was simple but efficient. Handover templates or automated solutions must be created in order to avoid information loss between handoffs.

4. Developing teams' trust before and during the project

Team members with different nationalities, experience levels and culture are susceptible to losing trust during the project. For example, some nationalities are very strict with the times and others have the opposite behavior. Such differences can easily have negative consequences for the project. Another example that can also cause misunderstandings between team members is the use of unknown terms in the software development context. Thus, we recommend training sessions about cultural aspects before and during the project. Using of such sessions for distributed teams is also cited by Setamanit et al. [7].

5. Ensure compliance of deadlines

Establishing deadlines at the beginning of the project is a usual practice in software development. However, constraints from local software development environments may result in delays in completing tasks, and consequently in delivering the final product. In order to ensure compliance with deadlines, events that can result in delays for projects must be controlled.

6. Establishing time rules

Handoffs tend to exceed the time duration [12], thus increasing the number of hours actually input to the project. In this study, we identified that the establishment of sometime rules contributed to a reduction in the duration of handoffs. The researchers created rules to ensure that time was shared amongst all team participants. The five first minutes were allocated for the Project Manager to discuss the new tasks, 10 minutes for CP givers, 10 minutes for CP receivers and 5 minutes to summarize the tasks gave by the CP givers.

7. Providing proper technologies and tools

In order to ensure that teams can perform all activities of the project within the estimated time, appropriate technologies and tools must be adopted in the project. There are many available technologies and tools supporting communication in distributed projects. However, the hardware compatibility between sites and the reliability of these technologies and tools must be evaluated before the project starts [23].

8. Delegating responsibility

At the beginning of the project, many tasks were not completed by the team. This occurred because the tasks assigned were not properly managed. A solution to ensure that tasks will be completed by the team is to delegate responsible members to each task. Each responsible member has to inform the Project Manager when his/her tasks were finished.

The Project Manager can use the information provide by responsible members to assign extra tasks, if it becomes necessary.

9. Coordination of task allocation

The approach adopted to task allocation in FTS environments can affect the efficiency in which handoffs will be performed by the team. Thus, the task allocation approach must support handoffs and vice-versa. Additionally, managers must be responsible for ensuring the visibility of tasks, team members, project structure and reporting structures across the global team [20].

6.3. Limitations and future research directions

This study has limitations. First, we studied handoffs management only in one software project.

Even though our evidence may not be generalized, there are many characteristics discussed in this study that are similar in other FTS projects.

Team members experience is also a limitation. Due to company constrains it was not possible to allocate team members with similar levels of working experience to the project. Training sessions were performed in order to reduce the imbalance in the level of experience. Additionally, a Scrum master was allocated to coach and ensure that teams were following Scrum practices.

Improvements identified during sprint 1 were applied to sprint 2. However, this study does not report results from sprint 2. We did not collect data from sprint 2 due to constraints within the company.

Future research may involve performing more studies about handoffs' management. Studies are required to promote efficient handoffs in FTS projects. We will investigate whether agile methodologies can help in better communication and coordination in handoffs' management. Managerial elements presented in this study can contribute to build a software process to perform handoffs. Tools to support handoffs activities will be also investigated in the next stages of this research.

7. Conclusions

In this study, we presented an experience report of handoff's management on a FTS software project. Our experience in developing this study shows that handoffs are difficult to manage mainly due to the disparity of team member experience and culture. Additionally, the lack of appropriate software practices for performing handoff activities decreases their efficiency.

Handoff's management must consider communication, coordination and cultural aspects in order to decrease challenges and improve its efficiency. We highlight in this study nine managerial elements that should be considered by managers in order to improve handoffs' efficiency. These elements are: *1) Improving communication quality, 2) Ensure knowledge transfer between team members and production sites, 3) Management of handoff's information, 4) Developing team's trust before and along the project, 5) Ensure compliance of deadlines, 6) Establishing time rules, 7) Providing proper technologies and tools, 8) Delegating responsibility,* and *9) Coordination of task allocation.*

In this study, we also investigated the duration of handoffs. If handoffs are efficiently managed, they can contribute to reduce software development duration. FTS is designed to reduce the duration of the software development cycle and its subprocesses, and should be performed in order to increase the speed. The nine managerial elements identified during this study help to reduce the duration.

8. Acknowledgment

This project was financially supported by the Infosys Technologies Company-India. The authors are grateful for the access and support provided by the Infosys Technologies Company involved in this study. The first author is supported by CAPES (Brazilian Ministry of Education) and the second author by 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. We also thank the PDTI program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91).

9. References

[1] Šmite, D., Moe, N. B., and Ågerfalk, P. J., Agility Across Time and Space: Implementing Agile Methods in Global Software Projects. 1st Edition, XXXVI, 341 p. 37 illus, 2010.

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

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

[4] N. Denny, S. Mani, R.S. Nadella, M. Swaminathan, and J. Sandal, "Hybrid Offshoring: Composite Personae and Evolving Collaboration Technologies", IRMJ 21 (1): 89-104, 2008.

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

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

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

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

[9] 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 International Conference on Global Software Engineering (ICGSE), 165- 174, 2010.

[10] P. Sooraj and P. K. J. Mohapatra, “Modeling the 24-h software development process”, Strategic Outsourcing: An International Journal, 122-141, 2008.

[11] 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 PARIS: Methods and Tools for Project/Architecture/Risk Management in Globally Distributed Software Development Projects, Proceedings of International Conference on Global Software Development (ICGSE), Bari, Italy, 2013.

[12] E. Hess and J. L. N. Audy, “FTSProc: a Process to Alleviate the Challenges of Projects that Use the Follow-the-Sun Strategy”, International Conference on Global Software Engineering (ICGSE), Porto Alegre, Brazil, 2012.

[13] G. Fadel, U. Lindemann, and R. Anderl, “Multi-National Around the Clock Collaborative Senior Design Project”, Invited paper, Honorable mention at the ASME Curriculum Innovations Award 2000.

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

[15] 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, 2002.

[16] 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, 2012.

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

[18] J. Kroll, R. Prikladnicki, J. L. N. Audy, E. Carmel, and J. Fernandez, “A Feasibility Study of Follow-the-Sun Software Development for GSD Projects”, 25th International Conference on Software Engineering & Knowledge (SEKE), 2013.

[19] Sommerville, I., Software Engineering, Ninth edition, Addison-Wesley, 792 pp., 2011.

[20] I. Richardson, S. Moore, D. Paulish, V. Casey, and D. Zage, “Globalizing Software Development in the Local Classroom”, 20th Conference on Software Engineering Education & Training (CSEET '07), pp.64,71, 2007.

[21] 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.

[22] E. O. Conchúir, H. Holmström, P. J. Ågerfalk, B. Fitzgerald, “Global Software Development: Never Mind the Problems – Are There Really Any Benefits?”, Proc. 29th Information Systems Research Seminar in Scandinavia, 2006.

[23] N. Denny, I. Crk, Igor, R. S. Nadella, and A. Gupta, “Agile Software Processes for the 24-Hour Knowledge Factory Environment”, (February 27, 2009). Available at SSRN: http://ssrn.com/abstract=1017184 or http://dx.doi.org/10.2139/ssrn.101718.