

# A Feasibility Study of Follow-the-Sun Software Development for GSD Projects

Josiane Kroll, Rafael Prikladnicki, Jorge L. N. Audy  
Computer Science School, PUCRS  
Porto Alegre, Brazil  
josi.unc@gmail.com, rafaelp@pucrs.br, audy@pucrs.br

Erran Carmel  
Kogod School of Business, AU  
Washington DC, USA  
carmel@american.edu

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

**Abstract**— Follow-the-sun (FTS) is a strategy for Global Software Development (GSD) where you hand off work at the end of every day from one site to the next, many time zones away, in order to speed up product development. Companies have tried to implement FTS, but have abandoned it after some point because of the difficulty to put it into practice. Consequently, there are few documented industry successes. The lack of FTS experience in the software industry is observed as the main barrier for its adoption. For this reason, we performed a study applying FTS to develop a software project. Our goal was to examine the feasibility and outcomes of FTS. In this paper, we present the experience report describing best practices and solutions performed to overcome the challenges we found.

**Index Terms**—Follow-the-sun, software engineering, global software development, coordination across time zones.

## I. INTRODUCTION

Follow-the-sun (FTS) is a software development strategy for Global Software Development (GSD) projects used to take advantage of temporal distances between several sites located in different time zones [1]. Its main goal is to reduce software development cycle duration [2]. However, FTS implementation requires great coordination, collaboration and communication with all team members involved [1].

While FTS concept looks promising in theory, it appears to be difficult into practice [3]. Many software companies have attempted to implement FTS, but have abandoned it after some point because of the difficulty of putting it into practice [1].

For this reason, this study aims to examine the FTS feasibility and understand the challenges and possible solutions for its development.

Our study was performed at Infosys Technologies in Bangalore, India in the second quarter of 2012. Over the duration of one month, working teams distributed in Mexico, India, and Australia developed a software application using FTS concept. In this study, we present our results, details of software practices and solutions performed to overcome the challenges found to develop a software application in the FTS mode. We also discuss feasibility issues, lessons learned and the next steps planned for this study.

## II. BACKGROUND FOLLOW-THE-SUN

In FTS scenarios, team members are distributed across different time zones and sites [2]. When team members from one site finishes its own regular working hours, other team members located in another site and time zone start its working

day. Tasks are handed off from one site to another at the end of each working day [4]. The tasks transition between team members is called a handoff.

Handoffs are performed daily by teams following to the next site. At each site, handoffs are conducted on a daily basis at the end of each site shift [5].

In the literature, FTS is also referenced as round-the-clock. Although these terms are used in a similar way in the literature, the definitions are different. FTS is about speed, cutting project duration, while round-the-clock is about twenty-four hour coverage, running an operation in all shifts. Both of these concepts use time zone differences to design shifts, but for different purposes and kinds of tasks [2].

## III. STUDY SETTINGS AND METHODS

Our research was developed at Infosys Technologies in Bangalore, India. Infosys is a global leader in consulting, technology and outsourcing with revenues of US\$ 7075 million (FY12). Infosys provides business consulting, technology, engineering and outsourcing services to help clients in over 32 countries.

### A. Case Setting

Our study focused in the development phase of a software application. This software application was developed by three distributed teams located in different time zones: Mexico, India and Australia.

In Mexico, there were two developers who were available full-time, in India two developers available for half-time and one project manager and in Australia we had two developers full-time and one developer half-time.

The sites had different experience levels. In Mexico, developers were trainees. In India, the project manager had working experience of approximately 10 years, did not have prior experience as a project manager. Developers from India, had two years and one year's experience respectively. In Australia, developers had between eight and fourteen years' experience.

### B. Project Planning

#### 1) Estimation

The application development was estimated to be completed in 4 weeks duration, and it was divided into two sprints. Since we did not have any estimation techniques or variations of standard techniques suited for FTS, we went more based on standard approaches and the experience of the project

manager. We also estimated the time for this project considering a typical two-location mode, which was estimated to be 6 weeks (this was done by an external experienced project manager not connected with this study).

2) *Task Allocation*

We followed the CPro concept introduced by [6]. Based on CPro formation concept, we formed 2 CPs (Composite Personas) with each CP comprising at least one team member from each location.

In each sprint, tasks were allocated to the CP, rather than to an individual. However, we found that this method caused some amount of confusion amongst CP members and resulted in some lack of direction and progress of the work in the sprint 1. Thus, this process was slightly modified in the sprint 2.

In the sprint 2, the project manager carried out twenty-four hour allocation of tasks to each CP. Following this each CP member would take on the tasks in the appropriate logical manner based on the tasks allocated for the twenty-four hours window.

3) *Team training*

We prepared guidelines for FTS teams based on the literature [1] [2]. These guidelines were reviewed by FTS experts before the project start. These guidelines were used to plan the project and conduct training sessions to clarify FTS.

Only the team from Australia had experience in agile methods. To mitigate this issue, we also conducted training sessions about the Scrum method. Additionally, a scrum master was allocated to the project and his role in the project was to coach and ensure that the teams followed Scrum.

C. *FTS Methods and Guidelines*

1) *Team setup*

Based on time zones where the teams were located and their available working hours, the daily working hours were arranged such that there was an overlap of 30 minutes between the locations for communication to enable the daily handover of tasks. Initially, the handover call was planned for 30 minutes in duration each day, subsequently it was found that it spilled over 45 minutes and 1 hour based on the need.

2) *Daily handover across locations*

Handoffs were performed over phone calls or communication tool. For sending tasks to another team, each developer should use an Excel template. This template was available on TFS system and it was called Task Handover. Individual developers were asked to add information for each handoff.

3) *Communication between team members across locations*

Communication between team members across locations was synchronous only during overlap times between the locations. A team member cannot talk with teammates from another site outside their time hours. Other forms of asynchronous communication were via email.

Time windows for interaction were available only one hour (maximum) per day. The first time window for interaction was available in the first 30 minutes (maximum) of a working day.

The second time window for interaction is available on the last 30 minutes (maximum) of a working day.

The working day started in Australia following to India and after to Mexico. Handoffs using conference calls between developers and project manager should be done every day to discuss new and performed tasks.

4) *Retrospectives at the end of each sprint*

Following the Scrum framework, at the end of sprint 1, a detailed retrospective was conducted giving opportunity for all members to voice out what they felt went right and what did not. This helped considerably to identify issues and also identify potential solutions and improvements to the overall process.

IV. RESULTS

A. *Performance*

1) *Performance in the sprint 1:* The tasks in the sprint 1 were estimated for an effort of 368 hours. At the end of sprint 1, it was found that effort expended was 432.5 hours, which was 65.5 hours more than planned. It was also found that only 65.5% of the planned tasks were completed with the rest incomplete. These tasks were moved to sprint 2.

One of the main hurdles encountered by the team was certain delays from the internal stakeholders which necessitated rework due to new templates introduced. This was estimated to have caused approximately 50% extra work. Similarly, the setup of the project took up more time than estimated. Finally, the daily FTS handover process also took more time than estimated, because planning meeting were executed during handoffs.

2) *Performance in the sprint 2:* Considering the existing tasks and the carryover tasks from sprint 1, the effort estimated for sprint 2 was 464 hours. In the sprint 2, we observed that teams were more comfortable and productive having getting experience in the FTS approach from sprint 1. Several of the problems faced in the sprint 1 were minimized in the sprint 2. The effort expended in this sprint was only 350 hours (see Table I) which was due to team members attending trainings and two holidays in one particular location. Consequently, the task completion was approximately 62% of the planned tasks.

TABLE I. PERFORMANCE IN THE SPRINT 1 AND SPRINT 2.

	<b>Sprint 1</b>	<b>Sprint 2</b>
Estimated hours	368	464
Actual hours	465.5	350
Extra hours	65.5	0
Task completion %	68%	62%

3) *Completion of the project:* the remaining tasks of the project were completed in a subsequent phase in non-FTS mode, because some team members were committed to other client projects and had to be released.

B. *Feasibility of the FTS model*

The core question this study sought to answer was: *Is FTS feasible in a live practical scenario in a large organization?*

From our experience in this project, our conclusion was that the FTS model is feasible. We present the main reasons next.

#### 1) *Work distribution and execution in round-the-clock manner*

This model was a major departure for all members of the team. The team went through some learning and ultimately settled down into a practical mode. At the overall level, the sprint user stories were divided amongst the two CPs. Further, within a CP, in the sprint 1, tasks were not specifically allocated to begin with. After facing some issues, this model was modified to some extent by adopting a 24 hours task allocation model, according to this, the project manager allocated tasks to each CP for each 24 hours. This practice brought in greater clarity in the team helping in better execution in the sprint 2.

#### 2) *Ownership of the CPro*

In the sprint 1, due to the lack of defined CP ownership, this ownership was taken by the project manager which was not very effective. Post the retrospective, this model was modified and specific owners for each CP was defined, thus ensuring more commitment and execution to plan.

#### 3) *Daily Handover Process*

Task handover from one time zone location to the CP member in the next time zone location was a concept tried out for the first time. As described above, and learning from previous research, a simple process was adopted. The team members reported a high level of comfort and satisfaction in the handover process and it was seen to be a good enabler in the FTS model.

### C. *Inherent Issues found in the Project*

#### 1) *Project-specific challenges*

A business application was chosen for this study with a defined end customer and stakeholders. While the end customer was supportive of the project, due to certain organizational constraints, the project faced considerable delays on account of other stakeholders. This contributed a certain amount of delay especially in the sprint 1.

The total lack of experience in one location and relative lesser experience in the second location, coupled with the higher experience in the third location, while, not entirely unusual, had some impact on the team working and productivity. Further, only one location members had adequate experience in agile software development, which meant the others had a learning curve on agile to be tackled in the project.

#### 2) *FTS methods and practices*

The biggest limitation experienced in this study was the lack of good estimation techniques for the project. The team used conventional estimation techniques to arrive at a target of 4 weeks for the project to be completed, whilst a neutral estimate of the same project for two location model pegged the estimate at 6 weeks. However, the project finally took 5.5 weeks to complete. As a consequence, the study did not show a significant cycle time reduction as it could, although FTS shows to be faster than a software project develop in non-FTS mode.

#### 3) *Communication and Coordination issues*

The main problem faced related to communication was language. While English was the mode of communication, due to differing accents spoken, team members across locations had some trouble understanding each other during the handoff meetings. Extra emails were requested by teams at the end of each handoff.

Coordination problems observed related to weekend handoffs task allocation and office timing management. Task allocation was problematic mainly at the beginning of the project, because tasks were not allocated properly. In addition, tasks not completed at the end of the day are handed to the next production site. Relate to office timing management, India team worked haft time and Australia team had a developer working half time.

### D. *Lessons Learned*

With the information collected during the project, we highlight some lessons learned.

- *Templates and standard document*: at the beginning of the project, teams faced problems to identify standards utilized in the project. Teams must know templates and standard documents that will be used during the software development before the project start.
- *Coding standards*: to avoid re-work a standard to comment code must be defined before the coding starts. When the FTS project started, teams were spending a lot of time trying to understand the code and identifying the last changes made in the code.
- *Screen sharing*: transferring or explaining a task using screen sharing becomes easy when teams can see the information talked about. During the handoff process we observed that teams opted by using of the screen sharing to explain codes and design documents.
- *Communication protocol*: we observed that phone calls, emails and communicators, such as Microsoft office communicator, are useful to provide communication between teams, but they must be used together. We also observed during the calls meeting that some rules following by teams can improve the quality communication, such as, speak slowly to reduce accents and summarize the tasks talked about by CP giver.
- *Tasks for the day*: a daily email allocating tasks individually for members contributed to define priority tasks and reduce problems faced by teams to categorize a task in the sprint backlog.
- *Handover template*: we used an excel spreadsheet to manage tasks exchanged between teams. It worked very well, but it could be automatized.
- *Weekend handoff*: in the weekends is very difficult to manage the handoff processes. In this project, we have used communication via email. However, many problems were identified mainly in the first weekend. The receiver team faced difficulties to understand the new tasks and how to continue the work. On the second weekend was better, but the tasks were discussed before starting the weekend.

- *CP owner*: some tasks were assigned to a CP owner during the sprint 1. We observed that is a good way to ensure complete tasks. Tasks can be assigned by email to CP owners per location. Each CP owner will check if the task has been completed.

## V. DISCUSSION

We consider some practices from literature designed for around-the-clock environments and it was adapted for FTS model. The experience of Infosys' experts allowed to improve practices and to create a software process for FTS. The adoption of Scrum practices was considered innovative in this context.

At the end of sprint 1, some changes were made in the process for the sprint 2. These changes are present as lessons learned in the section D.

We observed that the imbalance experience level had a negative effect on the project. The lack of experience affected from the project level to estimate the hours to complete tasks as well as the execution of tasks. Other challenges were identified like task allocation and lacking of standards and templates at the beginning of the project. However, these challenges were minimized for the next sprint.

### A. Constraints & Limitations

This study has some limitations that must consider:

- *Imbalance team's experience*: employees with different experience levels were allocated for this study. To minimize possible threats, we conducted training sessions with entire team before study start. In addition, we created guidelines giving instructions about FTS approach and scrum methodology.
- *FTS experience*: the software process followed by FTS teams was created by Infosys experts based on own methodologies. Experts in FTS and agile methods reviewed guides, practices and processes.
- *Agile experience*: the lack of the team's experience in agile was minimized allocating one scrum master for the project. His role in the project was to make the FTS team follow scrum method.
- *Single application*: one of the major limitations of this study is that they have examined only a single system developed by a single organization.
- *Team availability*: in the middle of sprint 2, team members from two locations were allocated to other client projects. For this reason, the project was completed using a non-FTS mode.

## VI. CONCLUSIONS AND FUTURE WORK

In this study, we reported the experience acquired with FTS applied to develop a software project. We used best practices from the literature and experts from the Infosys to create a process for FTS. Many software practices performed shown to be effective for FTS. In other hand, others shown to be ineffective resulting rework hours.

The main contribution of this study relates to the feasibility of FTS. Our findings show that FTS works for GSD projects

with some evidence that FTS can be used to compress duration. However, many untypical issues had occurred during the project. Team members attending trainings and developers without experience allocated to the project, are some examples. In the end, the take away from this study at Infosys is that FTS is feasible.

### A. Future Work

First, FTS needs to be experimented with more projects. Our study has shown good results using Scrum practices, but there is a need to gain more experience and understanding of when it works well, and how making it work better.

Another future opportunity is to study the impact of FTS in software projects in terms of cycle time reduction, as this is the main benefit expected from the implementation of a FTS project.

Our findings show that FTS is feasible, but it is hard to execute. We observed that few studies in the literature report solutions regarding to team coordination, task allocation and the process for daily handoffs.

Finally, the experience at Infosys show that FTS is an alternative to develop global software projects spread out in different time zones. Learning how to take the advantages for applying FTS successfully its part of the next steps at Infosys.

## 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. We also thank the PDTI program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91) and Lero - The Irish Software Engineering Research Centre ([www.lero.ie](http://www.lero.ie)). Rafael Prikladnicki is a CNPq researcher (309000/2012-2).

## REFERENCES

- [1] E. Carmel, J. Espinosa and Y. Dubinsky, "Follow the Sun Workflow in Global Software Development," *Journal of Management Information Systems* Vol. 27 No. 1, 2010, 17 – 38.
- [2] E. Carmel and J. A. Espinosa, *I'm working While They're Sleeping: Time Zone Separation Challenges and Solutions*, Kindle Edition, 2011, 188 p.
- [3] 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.
- [4] 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.
- [5] E. Hess and J. L. N. Audy, "FTSPROC: a Process to Alleviate the Challenges of Projects that Use the Follow-the-Sun Strategy," *In: 7th International Conference on Global Software Engineering (ICGSE)*, 2012, Porto Alegre, Brazil.
- [6] A. Gupta, R. Bondade and N. Denny, "Software Development Using the 24-Hour Knowledge Factory Paradigm," (April 29, 2008). Available at SSRN: <http://ssrn.com/abstract=1130062>.