2011 Sixth IEEE International Conference on Global Software Engineering

Researching into Follow-the-Sun Software Development: Challenges and Opportunities

Josiane Kroll, Estevão Ricardo Hess, Jorge Luis Nicolas Audy, Rafael Prikladnicki
Computer Science School
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre, Brazil
josiane.kroll@acad.pucrs.br, estevão.hess@acad.pucs.br, audy@pucrs.br, rafaelp@pucrs.br

Abstract — **Several organizations that develop software in global scale face issues related to coordination, communication and culture. The Follow-the-Sun (FTS) strategy is a type of global development with the purpose of reducing the duration of the project development and hence, increases productivity. However, FTS is not easy and it is not well discussed in the literature, with few studies discussing challenges and research opportunities. In order to promote a better understanding of FTS, this paper presents the state of the art about FTS, as well as the challenges and research opportunities, helping others in the planning of future studies in this area.**

Keywords - Global Software Development; Follow-the-Sun; Software Engineering.

I. INTRODUCTION

Global software development (GSD) is becoming a trend for companies that aim to keep competitive in the software development industry. GSD is also referenced as Global Software Engineering (GSE) or Distributed Software Development (DSD), and can be defined as software development with teams distributed among different geographical locations [1]. GSD is characterized when one or more individuals involved in the project are physically distant from another [2]. In GSD, one of the main characteristics is the time zone differences between production sites [3].

According to several studies [4, 5, 6] the time zone difference is difficult to manage. However, the time difference can also be used as an advantage and not only as a disadvantage [4, 6, 7, 8, 9, 10]. In this context, emerge the concept of Follow-the-Sun (FTS) software development. The FTS approach utilizes distributed team members spread across time-zones to achieve a single project outcome [11]. The main objective of FTS is to gain speed, reducing the time-to-market of a project [12].

FTS is an important research area relatively understudied within Software Engineering [7]. The success cases in the industry using FTS are insufficient [10]. Carmel, Espinosa and Dubinsky [7] claim that there is few documented success cases in industry. Thus, aiming to promote a greater understanding of FTS, this paper presents the FTS theoretical background and the contextualization of this area, highlighting challenges and research opportunities. The contributions of this paper are related to the FTS definition based on the perspective of different authors, and the identification of challenges and opportunities related to communication, coordination and culture in the context of FTS. We also present research opportunities to be explored in the future.

This paper is structured as follow: Section 2 presents the state-of-art of FTS. In Section 3 we present the challenges associated with the FTS. In the Section 4, we present research opportunities. Finally, Section 5 presents the conclusions.

II. FOLLOW- THE-SUN'S STATE-OF-ART

A. Theoretical Background

The times zones differences and geographical diversity present in GSD environment encouraged the development of innovative techniques and strategies that can move us towards the 100% utilization of the day [11]. FTS is one of these strategies.

The first documented experience of FTS usage happened in 1997 [13]. In this situation, IBM decided to develop a project using FTS. To achieve this goal, they created five teams divided into five distinct development centers, and in five different countries. During the project development, many coordination problems were found, especially during the daily handoffs. Because the FTS strategy was not bringing the expected results and several problems were being faced, the responsible for the project dropped out the usage of FTS to accelerate the development process, keeping only the GSD strategy.

Treinen and Miller-Frost [6] give details of two additional case studies at IBM. Their first case study describes a software project involving development sites in the United States and Australia. In this case study, two geographically distant development groups were merged into one cohesive team for FTS development. This project was considered a highly success. The second case study involved three distinct projects with sites in the United States and India. The three projects in this case study attempted a true FTS approach where several different developers were working on the same code base. Unfortunately, because the design team had never worked with the remote team in India, the specifications in design phase were often based on assumptions that were easily misinterpreted by the remote team. In addition, due to time constraints, the local team would often simply take on the

978-0-7695-4503-5/11 $26.00 © 2011 IEEE
DOI 10.1109/ICGSE.2011.31

rework and development themselves, regardless of project budget resulting in several problems. These problems and several challenges related to time-zone issues, different configurations of the development environment, cost estimates for the project and, team's cultural differences contributed for the failure of these three projects.

Carmel [14] also detailed the use of FTS by Infosys in 2005. Infosys did not apply FTS, but they practiced it in some project phases. The FTS was applied by a day, several days or in several months during the project. Infosys believed that FTS is limited and highly dependent on the type of task and the phases of the software development life cycle.

Other studies such as Gorton et al. [15], Lings et al. [8], Ramesh and Dennis [16] also documented the FTS usage. However, according to Visser [17], the FTS concept is not widely used in industry, because the number of documented case studies is limited.

Studies such as Holmstrom et al. [4] and Jalote and Jain [18] address the use of FTS by organizations. In all these studies we can observe that FTS has evolved in terms of concepts and definitions, but there is still a lack of studies published in the literature documenting successful stories of FTS implementation [19].

B. FTS Definition

In the literature, the term FTS is also referenced as round-the-clock [7, 17] and around-the-clock [19, 17]. Although these terms are used in a similar way in the literature, the definitions are very different.

FTS development referrers to unfinished work, usually software related, being handoff from one site to the next, generally on a daily basis. The round-the-clock and around-the-clock are related to the 24-hour coverage, developing a given project in all shifts. In this study we used the term Follow-the-Sun (FTS), because our focus is in cutting project duration [7, 12].

There are several definitions of FTS published in the literature. However, these definitions concentrate mainly in five studies [6, 7, 17, 20, 21]. Gorton and Motwani [20], for example, define that FTS development is related to software shift work. The authors show that the 24 hours development can be achieved only using the time difference among the sites where teams are allocated. Each team should work on its own regular work time and, at the end of its shift (its workday) the task is handed off to the next team, which will be starting its own shift and will take the work where the previous team left off.

Treinen and Miller-Frost [6] define FTS in a more succinct manner. For these authors, the time difference should be seen as an advantage to distribute the teams. The goal is to create a software development environment where teams work only during their regular working hours, and at the end of the day, only reassign their tasks to the team that is starting its work, creating effectively the 24-hour development.

Visser [17] defines FTS as a kind of software development, which can be determined by having software development teams spread across multiple time zones. Each team, at the end of the workday, delivers the relevant information regarding the work done so far, along with the product source code to the next team, which is starting its workday. The work will be continued from where the previous team left off. Thus, the project is being developed for 24-hours continuously and not only for eight hours, as it is conventionally done.

As for Gupta et al [21], FTS development can be extended to different tasks, besides the development. The authors advocate the knowledge factory concept, where the tasks to be developed in the project may be related to the knowledge, which is handed off among globally distributed teams at the end of each day. To illustrate this concept for the 24 hours continuous development, they presented the testing phase, where the knowledge is the software itself, passed among the development and test teams, which were distributed in different sites.

Carmel, Espinosa, and Dubinsky [7, 12], defines FTS development as a global knowledge workflow, that aim to reduce a project duration, in which the product is owned by a production site until it is handed-off to the next team. This handoff should be done in a daily basis and this next team must be allocated many time-zones apart.

Based on the definitions given by different authors, a comparative analysis among all the definitions was developed. For each criterion (columns), we aimed to know whether the given definition, in some way, indicated the importance of this factor. We considered important factors those that define specific characteristic or aspect of the FTS development.

The factors used for comparative analysis arose from a prior assessment of the presented concepts. We aimed to identify the key factors for each concept in the literature, which could be compared among all definitions. Table 1 shows this comparison.

TABLE I. COMPARISON AMONG DEFINITIONS OF FTS

Reference	***FTS main objective***			***Daily handoff***		***Considers relevant***	
	Increase speed of the project development	Reduce the time to market	Increase the daily development time	Source code (Coding phase only)	Extended to other tasks	Work only in the regular time	Time difference of many time zones
Visser (2009) [17]			X	X	X		X
Gorton and Motwani (1996) [20]	X	X		X	X	X	
Gupta et al. (2009) [21]	X		X		X		
Treinen and Miller-Frost (2006) [6]				X	X	X	
Carmel, Dubinsky and Espinosa (2009) [7, 12]	X	X		X		X	X

As one can observe in Table 1, there is no convergence among the definitions for the FTS. However, the definitions found complement each other, with specifically characteristics that contribute to better define FTS.

The definitions related to the primary purpose of the usage of FTS, although expressed differently by the authors, define that the main goal of FTS is to increase speed to develop a project thus, reduce the time-to-market. The minimum difference time zones required across teams were underreported. There are also few studies that consider relevant the importance to have different teams working on their regular day period to avoid overtime at night.

Therefore, after the comparative analysis of the definitions suggested by different authors, relevant points were identified in each definition. Based on the information collected, we list a set of characteristics that describe FTS, as following:

- FTS is a global software development strategy;
- The main goal is to decrease the time-to-market thus, speeding up the product development;
- This environment works with teams spread in different countries and time zones;
- Each team has to work for a specified time slice;
- The handoff can be applied to any software development task;
- This handoff should occur on a daily basis and in a standardized way.

C. *Related Work*

In the literature there are few studies exploring FTS. The published studies are mainly focused on the characteristics and approaches to use FTS. An example is the work presented by Treinen and Miller-Frost [6]. The authors use cases study to evaluate the FTS development in GSD environments. This study analyzed the factors that influence the project's success or failure and discussed the best practices for successfully usage of the FTS approach.

Setamanit, Wakeland e Raffo [9] showed the advantage for a project that uses the FTS strategy over a project developed in a single site. The objective was to identify when there is an advantage by using FTS, and what are the requirements to achieve an advantage that is interesting for a project developed using this strategy. This study reported from a software development project, comparing a traditional project with teams collocated and a project that was executed in a GSD environment, using FTS. During this study, the authors used two geographical distributions for the team that used the FTS strategy. The first one was done with two teams, with none overlap between them. In this situation, the project length increased 50%. The authors say that this is related to the communication issues during the handoffs. In the second distribution, they used three teams, using an overlap between them. This situation brought a better result: the project time was 11% less than the collocated team. At the end of this study, the authors state that FTS can be a good strategy if the project team wants to focus on time reduction. Nevertheless, they also emphasize that if there is the decision to use the FTS strategy the team distribution should be with at least three development teams, with an overlap between them. This overlap period would be used to have synchronous communication among the teams.

Finally, other study published by Carmel, Dubinsky and Espinosa [7] address the FTS aiming at to identify new perspectives and characteristics. The authors presented a definition for FTS development, and described phases from the software development life cycle that best fits the FTS strategy. At the end of the paper, the authors concluded that teams that use FTS finish their work faster than collocated teams. The time gain was 10%, but as per the authors, this gain could be even higher.

III. FTS CHALLENGES

In GSD, apart from being geographically dispersed, globally dispersed teams face challenges related to time zone differences, and social, cultural and normative differences. These differences in time zone and cultural background add challenges to collocated software development, mainly related to communication, coordination, and culture [1]. FTS is a subset of GSD and have the same challenges [7, 15, 20, 22]. According to Carmel, Dubinsky and Espinosa [12], FTS has challenges such as coordination barriers, cultural differences, and communication difficulties.

The coordination barriers challenges are mainly associated to the number of sites that are part of a certain project [9, 10]. When more than one site is added to the project, this increases the difficulties to coordinate aspects that involve team management, and cultural and geographical differences [7]. For example, the coordination of the continuity of work involves daily handoff cycles among teams. The handoffs are difficult to coordinate due to the difficulty of resolving task issues across sites/shifts, and the cross-site coordination cost will most likely be positive and nontrivial. Unfortunately, we noticed few successful cases of FTS, and one of the reasons for that is the challenge related to coordination [7].

The increase of the number of sites also adds difficulties to communication. These difficulties occur due to the increasing of the number of teams allocated to the project and consequently loss of communication richness [7]. The communication challenges in FTS are associated mainly to the lack of synchronous communication between distributed teams [9].

Culture differences challenges are associated to the socio-cultural diversity present in FTS development environments. It is determined mainly by social, ethnic and cultural aspects [23, 1). Holmstrom et al. [4] argue that when constraints, such as temporal, geographical and socio-cultural distance are identified, and while they increase in the scope of organizational operation, these constraints result in challenges for FTS. For example, the usual problems of supporting collaboration are compounded by language and diversity [11]. In addition, if one culture has more emphasis on self sufficiency, therefore they tend not to ask for help when problems come up. Another culture would not offer their help

unless they were asked while the third considered that presenting the problem was a sufficient invitation for willing team members to jump in and help. According to Yap [19] the cultural differences often created misunderstandings and lead to frustration and conflicts between teams.

FTS challenges restrict it application by organizations. Therefore, the challenges have to be minimized in order to work effectively across geographical, cultural and temporal boundaries [11]. The implementation of FTS, if not correct, may result in failures and over budget projects [24].

IV. RESEARCH OPPORTUNITIES FOR FTS

FTS is a research area with many important aspects to be investigated [21]. Based on the challenges inherent to the FTS strategy, it is important to conduct studies in order to collect evidences that show practices, pitfalls, and ways to have successful implementations of FTS.

We identified twelve research opportunities related to FTS in peer-reviewed literature. Each research opportunity can be associated to more than one challenge and all of them are presented in Table 2.

TABLE II. FTS' RESEARCH OPPORTUNITIES

Research Opportunity	Challenge	References
1. FTS adaption to different phases of the software development life cycle	Coordination	[12]
2. Issues of scheduling tasks to individuals resources for reducing the overall execution time of a project		[18]
3. Team building for FTS		[7, 25]
4. Coordination costs in FTS		[12]
5. Calendar efficiency		[12]
6. Daily handoff cycles		[7, 10, 12, 26]
7. Reduction of communication problems	Communication	[21]
8. Knowledge sharing practices and knowledge reuse	Coordination / Communication	[9, 21, 27]
9. Development method and methodologies	Coordination / Communication / Culture	[12, 19]
10. Efficacy of coordination mechanisms in the context of software development		[1, 27]
11. Time zone overlap		[12, 24]
12. Reduction of the effects of cultural diversity	Culture	[7, 9, 11, 23, 24]

Each research opportunity presented is detailed next:

1. *FTS adaption to different phases of the software development life cycle*

The research opportunities related to coordination challenges represent mainly the need for managing the software development that involve resources (e.g., human and physical), time and tasks. According to Carmel, Espinosa and, Dubinsky [12], the coordination is complicated when working in distributed environments because it entails both managerial aspects (e.g., coordinate per customer requirements) as well as technical aspects (e.g., coordinate per architecture decisions). The *FTS adaption to different phases of the software development life cycle* is a research opportunity, once each phase has its own specifics characteristics and the FTS adoption might change from one phase to another [12].

2. *Issues of scheduling tasks to individuals for reducing the overall execution time of a project*

Jalote and Jain [18] consider that to harness the potential of the FTS for reducing the overall development time, a key issue is the allocation of project tasks to the individuals in the distributed team. However, it is observed that when a team works on 24-h cycle, the efficiency is reduced by the overhead involved [21]. This way, researches about issues of scheduling of tasks to individuals for reducing the overall execution time of a project are very important to alleviate problems of coordination.

3. *Team building for FTS*

The *Team building* is also a coordination challenge, since in FTS development, to build and maintain cohesiveness in dispersed teams is difficult [1]. According to Lindemann, Anderl and Gierhardt [25] in addition to the other elements of GSD, the process of forming dispersed team is a significant challenge. Studies can explore this challenge and develop strategies to identify the characteristics that a team should have to be part of a FTS strategy.

4. *Coordination costs in FTS*

Carmel, Espinosa and, Dubisnky [12] say that one of the unresolved FTS issue is *Coordination costs.* The highly interdependent work that FTS imposes has different coordination costs relative to both co-located and conventional global work configurations. Thus, optimizing coordination costs in the FTS strategy is important to reduce time-to-market.

5. *Calendar efficiency*

The main benefit of the 24-hour development is the speeding up of work [28]. In the literature studies have devoted attention to investigating the time domain but have largely focused on perceptions of time [12]. These studies seek to improve the calendar efficiency and consequently time-to-market reduction. However, there is a lack of studies related to how calendar efficiency can help to analyze, design and implement successful FTS practices that can reduce task duration [7].

6. *Daily handoff cycles*

A handoff is defined as a check-in of a work unit that the next site is dependent upon in order to continue that work [7]. The handoffs are performed daily among the production sites. In this scenario, developers have difficulty in predicting their own productivity. This difficulty persists when attempting to estimate the productivity of others [26]. An IBM team described in Carmel's book [13] decided to abandon the FTS strategy because the daily handoffs among sites were too difficult to coordinate. According to Setamanit, Wakeland, and

Raffo [9] if the handoffs could be made smoother and faster, then cycle times could be reduced. Moreover, if coordination could be made more efficient, then cycle times could be reduced.

7. Reduction of communication problems

Geographic distance and time zones difference reduces the opportunity for face-to-face interaction [21], losing the communication richness [7]. For this reason, there are significant communication difficulties in FTS. Communication in FTS can be either synchronous or asynchronous, depending on the time zone overlap among the production sites and the office hours. In addition, if communication is not properly addressed, it can create problems that need to be solver or at least minimized.

8. Knowledge sharing practices and knowledge reuse

Effective knowledge sharing is considered essential for high performance in both collocated and distributed teams [21]. However, in FTS development the distributed teams may be missing key information that would help them to function effectively. This research opportunity is associated to communication and coordination challenges, because distributed team requires more handoffs of knowledge, and thus requires more formal systems to facilitate the coordination of these handoffs.

9. Development method and methodologies

According to Carmel, Espinosa and, Dubisnky [12], there is a lack of methods to drive FTS implementation. These methods have to be associated with coordination, communication and culture challenges. The authors state that, for example, FTS is more suitable for agile development and this should be better investigated in future research.

10. Efficacy of coordination mechanisms in the context of software development

The *Efficacy of coordination mechanisms in the context of software development* is a research opportunity that has many opportunities to be studied. For example, mechanisms for transferring work-in-progress to the next production site in a timely manner, efficiently, and low overhead; mechanisms for knowledge transfer, estimate delivery schedules, reduce defects, among others [27].

11. Time zone overlap

Time-zone effectiveness implies in teams located across several time zones with, ideally, some overlapping work-hours during the day. However, sufficient overlap in working hours may be difficult to achieve [24]. According to Setamanit, Wakeland, and Raffo [9] the increasing in the overlap of work hours contributes for less development time per day. The time zone overlap involves coordination, communication and culture challenges due to the fact that different cultures make it more difficult to communicate and coordinate, which could result in longer project duration.

12. Reduction of the effects of cultural diversity

Several studies indicated that cultural diversity is one of the challenges of FTS [7, 9, 11, 23, 24]. Some companies such as HP and Intel reported various problems due to cultural differences within their FTS teams [24]. The cultural differences also are cited as problematic, especially with respect to various assumptions of requirements and planning [26]. This way, there are research opportunities in areas such as how to cope with different languages, laws, legislation, and socio-economic and cultural aspects.

V. CONCLUSIONS

Researching into FTS is important for GSD evolution. Several companies are looking for increasing their efficiency in software development and FTS could be an opportunity for them. However, we have few studies exploring FTS as a research area and presenting success cases. In this paper we presented the FTS theoretical background and the contextualization of this area, highlighting the opportunities in this research field.

The results presented showed that the challenges surrounding the FTS are mainly concentrated on issues involving coordination, communication and culture. FTS challenges are associated one with others. The reduction of a challenge may contribute to minimize other.

Few studies in this area have been published so far; however, there is a great opportunity to companies from all over the world. On increasing the research in this area we might contribute to the FTS adoption by these companies. For this reason, we aim to develop future studies in this area. We also identify twelve research opportunities to be investigated. It involves issues of coordination, communication and culture. Each research opportunity has several aspects that can be explored and involves key aspects of FTS. It is important to notice that this is not a restricted list, but some of the opportunities to be investigated.

ACKNOWLEDGMENT

The authors have a research grant from the PDTI program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91). The fourth author has also a research grant from Ci&T.

REFERENCES

[1] R. Jabangwe, I. Nurdiani., “Global Software Development Challenges and Mitigation Strategies: A Systematic Review and Survey Results,” Master´s program in Software Engineering, Blekinge Institute of Technology, OM/School of Computing, 2010.

[2] R. Prikladnicki, J. L. N. Audy, “Process models in the practice of distributed software development: A systematic review of the literature,”. Inf. Softw. Technol. 52, 8 (August 2010), 2010, 779-791.

[3] M. Lane, P. Ågerfalk , “Suitability of Particular Software Development Roles to Global Software Development,” 3rd IEEE International Conference on Global Software Engineering, 2008.

[4] H. Holmstrom, E. O. Conchuir, P. J. Agerfalk, 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,2006, pp. 3-11.

[5] J. D. Herbsleb, R. E. Grinter,“Architectures, coordination, and distance: Conway's law and beyond. Software,” IEEE , vol.16, no.5, 1999, pp. 63-70.

[6] J. J. Treinen, S. L. Miller-Frost, “Following the Sun: Case Studies in Global Software Development,” IBM Systems Journal, Volume 45,,Number 4, 2006.

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

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

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

[10] R. V. Solingen, 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), 2010 5th IEEE International Conference on , pp.165-174.

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

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

[13] E. Carmel, “Global Software Teams”, Prentice Hall,PTR: Upper Saddle River, NJ, 1999.

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

[15] I. Gorton, I. Hawryszkiewycz, L. Fung, “Enabling software shift work with groupware: a case study,” System Sciences, Proceedings of the Twenty-Ninth Hawaii International Conference on, vol.3, 1996, pp.72-81.

[16] V. Ramesh, 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.

[17] C. Visser, “Connecting Time Zones and Languages in Follow-the-Sun: a routing model,” Masther Thesis - Delf University of Technology – Holanda,2009.

[18] P. Jalote, G. Jain, “Assigning tasks in a 24-hour software development model,” Software Engineering Conference, 2004. 11th Asia-Pacific, pp. 309- 315.

[19] M. Yap, “Follow the sun: distributed extreme programming development,” Agile Development Conference (ADC'05),2005, pp. 218-224.

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

[21] A. Gupta, E. Mattarelli, S. Seshasai, 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, 2009, pp. 147-161.

[22] A. Gupta,“Deriving mutual benefits from offshore outsourcing,” Communications of the ACM, v.52 n.6, 2009.

[23] M. Jiménez, M. Piattini, A. Vizcaíno, “Challenges and improvements in distributed software development: A systematic review,” Advances in Software Engineering, 2009.

[24] E. Ó 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.

[25] U. Lindemann, R. Anderl, H. Gierhardt, G. M. Fadel, “24 hr Design and Development – An Engine Design Project,” In: Scrivener, S. A. R.; Ball, L. J.; Woodcock, A.: Collaborative Design, Chapter B. Springer, New York, 2000.

[26] N. Denny, I. Crk, R. Seshu, “Agile Software Processes for the 24-Hour Knowledge Factory Environment,” Journal of Information Technology Research, Vol. 1, Issue 1, pgs 57-71, Jan 2008

[27] A. Gupta, S. Seshasai, “24-Hour Knowledge Factory: Using Internet Technology to Leverage Spatial and Temporal Separations,” ACM Transactions on Internet Technology, Vol. 7, No. 3, Article 14, Aug 2007.

[28] A. Gupta, S. Seshasai, R. Aron, "Research Commentary: Toward the 24-Hour Knowledge Factory - A Prognosis of Practice and a Call for Concerted Research," (November 19, 2006). Eller College Management Working Paper No. 1038-06, 2006.