An Initial Framework for Researching Follow-the-Sun Software Development

Josiane Kroll, Bernardo José da Silva Estácio, Jorge L. N. Audy, Rafael Prikladnicki
Computer Science School
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Porto Alegre, Brazil
josi.unc@gmail.com, bernardo.estacio@acad.pucrs.br, audy@pucrs.br, rafaelp@pucrs.br

***Abstract*—Global software companies are restructuring their IT area by extending operations to offshore software development centers. Thus, follow-the-sun (FTS) development is seen as a potential software development strategy for these companies. But so far, FTS is relatively understudied (only nine empirical studies published). Moreover, the success cases of FTS usage in the software industry are still small. The lack of studies to close the gap between theory and practice is observed as the main barrier to the FTS evolution. In this study, we present a framework to support research on FTS software development. The proposed framework aims to support empirical studies and meta-analysis on FTS. Based on a comprehensive literature review, we propose an initial framework for researching FTS software development, including independent, dependent and context variables. Additionally, we discuss research challenges, and directions for future research. Our paper aims to provide both a guide and motivation for researchers to better understand how to research FTS and to develop new theories in the area.**

Keywords—follow-the-sun; empirical software engineering; global software development; research method; research variable; research challenges.

I. INTRODUCTION

Follow-the-sun (FTS) concept is an alternative for Global Software Development (GSD) environments when trying to manage problems related to temporal distance. FTS aims to reduce the total time of the project while working 24 hours across the time zone [1].

In FTS, team members are geographically distributed in different time zones. Team members work during their normal working hours. FTS includes a daily handoff of unfinished work from site to site [1].

FTS has always been enticing for companies, even risky [2]. Many companies, such as IBM and Infosys have tried to apply FTS, but abandoned it after some point, because of the difficulty of putting it into practice [3]. Today, Brazilian firms, like Indian firms, are looking to try FTS. In 2010, Politec (a large Brazilian IT firm) and Mitsubishi publicized their initiative targeted at Japanese companies to leverage FTS [2].

While software companies have been increasing its interest in FTS, in the literature is observed the lack of studies to develop theories and solutions for FTS implementation. According to Carmel, Espinosa and, Dubinsky [4], few studies have explored FTS development and there is little evidence of success. The lack of studies to close the gap between theory and practice is observed as the main barrier to the FTS evolution in the software industry [5].

FTS development requires more communication and coordination between sites [6]. Thus, together with temporal, cultural and language differences, few GSD projects are able to realize the full (theoretical) benefit of FTS [3]. Furthermore, software companies consider FTS difficult to achieve its benefits, because FTS requires a great effort from all the team. Additionally, if FTS is not properly applied, it can result in failures and increase the project costs [7].

In this paper, we present an initial framework for researching FTS software development, in order to support empirical studies and meta-analysis. The proposed framework categorizes independent, dependent and context variables of empirical studies. We also discuss research challenges and directions for future research.

The focus of the paper is to promote a better understanding on how to research FTS software development. Our motivation to present this paper is the lack of studies to close the gap between theory and practice.

Our study offers the following main contributions:

- **The initial framework for research on FTS:** since the proposed framework presents variables at various levels, it provides to researchers the opportunities to investigate FTS from different perspectives.
- **The theoretical foundation for meta-analysis**: the proposed framework serves as the foundation for future research and for combining the results of individual studies.

The remainder of this paper is organized as follows. In the next section, we present the literature review. In section 3, we present the proposed framework for research on FTS. In section 4, we discuss research challenges, and directions for future research. We draw our conclusions and future work in section 5.

978-1-4799-4360-9/14 $31.00 © 2014 IEEE
DOI 10.1109/ICGSE.2014.21

II. Literature Review

In this section, we introduce a background on FTS, focusing on aspects related to the proposed framework. First, we introduce the FTS concept. We also describe the main empirical studies on FTS and identify research variables (shown in Table 1). In order to identify research variables, we followed the definition given by Wholin [8] and Gallis, Arisholm and Dyba [9] (see Section 3). Then, we identify the relationship between empirical studies and research variables.

A. Follow-the-Sun Software Development

Follow-the-sun (FTS) is a subset of GSD. 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 [5]. Its main purpose is to reduce the software development life cycle duration or time-to-market [1].

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 [4].

As team members are distributed across multiple time zones, organizations can develop software twenty-four hours continuously. Thus, the time reduction may be theoretically by 50% if there are two sites and by 67% if there are three sites [10]. However, when the number of sites in a daily cycle increases, on average, the overall working speed of the sites also increases [5].

At the beginning and at the end of each working day shift there is a handoff. Handoff is a term adopted in the literature to define the process transition from one site to another [1]. 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 [3].

Performing handoffs creates dependencies between production sites [4]. 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 to implement FTS projects [1].

In the literature, FTS is also referenced as around-the-clock. Although these terms are used in a similar way, their definitions are different. FTS is about speed, cutting off project duration, while around-the-clock is about twenty-four hour coverage, running an operation in all shifts [1].

In around-the-clock development, there are teams with specific skills at each site. It occurs because teams located in different sites work on different tasks. In around-the-clock, each team is responsible for developing individual tasks and not sharing it with other sites. Around-the-clock concept is recommended for all software life cycle phases. On the other hand, FTS is recommended only for the development and the testing phases [4]. FTS can be applied to other phases, but it may be not result in time reduction or its application becomes difficult.

B. Empirical Studies on Follow-the-Sun

FTS is relatively understudied in Software Engineering [4]. Thus, in order to understand the existing research directions on FTS, this section summarizes the main empirical studies on FTS (findings based on direct evidence or experiment). These studies were identified by Kroll et al. [10]. Kroll et al. [10] conducted a Systematic Literature Review to identify best practices and challenges for FTS implementation. They substantially extend the empirical evaluation of FTS. Table 1 gives an overview of these studies.

In 1999, Carmel reports the first experience using FTS in the software industry [11]. In this situation, IBM decided to develop a project using FTS. In this project, five teams were created into five distinct development centers, and in five different countries. In this project, IBM faced many coordination problems, especially during daily handoffs. Because FTS was not bringing the expected results and several problems were being faced, those responsible for the project dropped off use of FTS to accelerate the development process, keeping only the GSD.

Yap [12] describes the experience using XP (Extreme Programming) to develop a FTS software project. In this study, the author reports challenges, lessons learned and solutions for global continuous integration such as, cultural differences and conflicts between sites. This study focuses mainly on XP distributed development using as a scenario FTS development.

Treinen and Miller-Frost [13] gave details of two additional case studies at IBM. The first case study describes a software project involving development sites in the US and Australia. In this case study, two geographically distant development teams were merged into one cohesive team for FTS development. This project was considered a success. The second case study involved three distinct projects with sites in the US and India. In these projects, teams worked on the same code base. Due to team's inexperience, time constraints and, project budget, the projects were dropped out. Several challenges related to time zone issues, different configurations of the development environment, project cost estimates and, team's cultural differences contributed to the failure of these three projects.

In 2007, Espinosa, Nan and, Carmel [14] conduct the first controlled experiment in FTS to investigate the impact of time zone overlap on speed and accuracy. In this study, 42 pairs of students were selected to simulate a FTS project. This study showed that development speed is higher at both full and zero overlap, but not in different fractions of overlap between sites. This study uses fictional maps to evaluate the impact of time zone overlap on speed and accuracy.

Carmel, Dubinsky and, Espinosa [15] performed a quasi-experiment to measure working speed in FTS between collocated and distributed teams. This study followed practices from agile methods. The findings of this study show that teams using FTS approach are faster than collocated teams. In this study, the time spent with the development increased by 10%, but for the authors, the time development increase could be even higher.

Kroll et al. [16] investigated the benefits from adaptive and prescriptive approaches for FTS development. This study also

TABLE I. EMPIRICAL STUDIES ON FTS

Ref.	Research method	Context variables						Independent variables	Dependent variables (metrics)
		Subjects	N. of subjects	Task	N. of tasks	Sites	Duration		
[11]	Case study	Professionals	Unknown	Software tasks	Unknown	Five distinct development centers in five different countries.	Unknown	Evaluation of FTS to develop software projects	Feasibility
[12]	Case study	Professionals	Unknown	Code base	Unknown	UK, US, Singapore	1 Year	Evaluation of XP to gather experience on its practices (methodology)	XP distributed development using a FTS scenario (feasibility)
[13]	Case study	Professionals	Unknown	Software tasks	Unknown	US, Australia	1 Year	Evaluation of FTS applied to develop software projects	Speed
[14]	Experiment	Students	84	Fictional maps	13	US	4 synthetic workdays (each workday with 15 minutes and 5 minute break)	Full overlap, 2/3 Overlap, 1/3 overlap and no overlap	Speed (number of maps a team could replicate divided by the total number of possible maps), accuracy (number of correct elements in the map and their correct position in the picture)
[10]	Quasi-experiment	Students	15	Software tasks (simulator of processes and thread scheduler in the Unix operating system)	1	Onsite	14 weeks	FTS team versus Control team	Speed (number of check-in operations and number of revisions per file per determined time)
[5]	Experiment	Students	3 or 4	Fictional maps	21	Simulation involving 4/3/2 sites	3 days	2 sites versus 3 sites versus 4 or more sites	Speed (number of elements created per day), individual speed (average amount of elements created per workday per development site), accuracy (number of correct elements)
[15]	Experiment	Students	12	Fictional maps	13	Onsite	1 hour and 15 minutes (five shifts of 15 minutes each)	Adaptive versus prescriptive approach	Speed (amount of maps delivered divide by the total number of maps), Accuracy (amount of points from correct elements in each map divide by the number of total points), Quality (sum of total accuracy points divided by the total number of maps delivered by each team)
[16]	Experiment	Students and Professionals	8	Software tasks (mathematical system)	12	Onsite	40 minutes (20 minutes for each shift)	FTSProc versus Adhoc	Efficiency (number of requirements implemented in a given time interval)
[17]	Case study	Professionals	5	An internal software application	1	India, Mexico, Australia	4 weeks	Evaluation of FTS to gather experience	Evaluation of feasibility and speed
[18]	Case study	Professionals	5	An internal software application	1	India, Mexico, Australia	4 weeks	Evaluation of handoffs to gather experience inon the process	Investigate how handoffs management should be performed in an FTS software development context (speed, quality)

uses fictional maps. Obtained results of this study have indicated that the usage of adaptive approaches increases the speed, but they do not always enhance accuracy and quality of the work done by distributed sites. In addition, based on the results found, the authors suggest that adaptive approaches could perform better than prescriptive approaches in the context of FTS.

Hess and Audy [17] propose a process to perform diary handoffs. This process aims to alleviate difficulties faced during the development phase of FTS software projects. A controlled experiment was performed to evaluate its efficiency. Evidences show that is possible to reduce development difficulties in FTS using the proposed process. The process is based on Composite Persona (CP) and 24hr Design and Development concepts. In addition, it uses Test-driven development (TDD) technique.

Recentelly Kroll et al. [18] conducted a case study at Infosys Technologies to examine the feasibility and outcomes of FTS. Over a month duration, working teams distributed in Mexico, India, and Australia developed a software application using FTS approach. This study presents details of software practices and solutions performed to overcome the challenges found to develop a software application in the FTS mode. The authors also discuss feasibility issues and lessons learned.

Kroll et al. [19] investigated how handoffs management should be performed in an FTS software development context. They present an experience report describing handoffs development and management in a FTS software project. The results describe the participants' perception about software engineering activities performed, challenges faced and solutions performed to minimize these challenges. They also highlight management elements for handoffs.

C. Relationship between empirical studies and research variables

Some studies are related to each others. Thus, in order to better understanding how variables remain relevant and useful to perform new studies, we identified related studies to variables (shown in Table 2, 3 and 4).

A related study in this paper means a previous study. We use the word '*Unknown*' to refer studies without a previous study or not identified. We organized studies in Table 2, 3 and 4 by year. Thus, a variable can appear more than one time in the same table. It means that such variable is discussed in more than one study. In addition, we list research topics related to variables (Column 3). In Table 3, an additional column, called *Specific variable*, is inserted. That column list one or more context variables. We use the word *'All'* to refer studies discussing all context variables.

Next, we describe these relationships between studies and variables.

- Independent variables

The feasibility of FTS was studied in first by Carmel [11]. In his study, Carmel reports many coordination problems and the project failure. Any study had discussed the feasibility before as shown in Table 2. Recentely, Kroll et al. [18] conducted another study to examine the feasibility of FTS. In this study, the authors make use of the solutions already proposed by previous studies as shown in Table 2. The authors conclude the study claiming FTS as feasible.

Agile methods are investigated in first by Yap [12] as shown Table 2. The author discusses the use of XP (Extreme Programming) to develop a globally distributed project.

Denny et al. [20] inspired in Yap [12], explored the utilization of agile practices for 24-Hour Knowledge Factory (24HrKF) environments. They aim to search for solutions that enable handoffs to be practiced effectively. Thus, this study describes the CPro process'. The core of CPro is a model of cooperative work called the Composite Persona (CP). Fadel, Lindemann and Anderl [21] introduce the CP concept. The study performed by Hess and Audy [17] is inspired in these studies.

Hess and Audy [17] propose a process for FTS based on studies performed by Denny et al. [19] and Fadel, Lindemann and Anderl [21]. Their process is called FTSProc. The process aims to mitigate the challenges of coordination, synchronization and communication during the handoffs in the development phase of SDLC (Software Development Life Cycle). Recentely, Kroll et al. [19] make use of the results of these studies to investigate how handoff's management should be performed.

Benefits of agile methods for FTS are investigated by Kroll et al. [16]. In 2010, Carmel, Espinosa and, Dubinsky [4] claimed the use of agile methods for FTS as promising practices. Yap [12] and Denny et al. [20] are related to Kroll et al. [16] as shown in Table 2. These studies have in common the variable methodology. Additionally, the use of agile methods for FTS was reported by Kroll et al. [18] and Kroll et al. [19]. These studies also adopt the CP concept discussed Denny et al. [20].

Overlap differences between sites [14] and the number of sites [5] are also investigated in FTS research. These independent variables does not have related studies as shown in Table 2.

TABLE II. INDEPENDENT VARIABLES AND RELATED STUDIES

Study	Independent Variable	Research topic	Year	Related study
[11]	Feasibility	FTS	1999	Unknown
[12]	Methodology	XP	2005	Unknown
[14]	Overlap difference	Time zone separation	2007	Unknown
[5]	Number of sites	Project setting	2010	Unknown
[17]	Process	Handoff	2012	[12] [20] [21]
[16]	Methodology	Adaptive versus Prescriptive	2012	[12] [20]
[18]	Feasibility	FTS	2013	[11] [12] [16] [20]
[19]	Process	Handoff	2014	[167] [20]

- Context variables

In the literature, we observed three types of context variables: subject, task and site variables.

Espinosa, Nan and, Carmel [14] conducted a controlled experiment with 42 pair of students. They also used fictional maps. Solingen and Valkema [5] and Kroll et al. [16] inspired in Espinosa, Nan and, Carmel [14] performed their study as shown in Table 3. These studies have subjects, tasks and site variables in common.

We observed the use of two types of tasks in FTS studies: software tasks and fictional maps. Software tasks performed by teams were different in studies. However, the fictional maps were the same.

Espinosa, Nan and, Carmel [14] simulate a FTS project using fictional maps at first. After that, Solingen and Valkema [5] and Kroll et al. [16] also adopted the same fictional maps. Studies having as a task, fictional maps, also shared the same context variables.

Kroll et al. [18] and Kroll et al. [19] conducted their studies inspired in the findings reported by Solingen and Valkema [5] as shown in Table 3. This study defines the number of sites for FTS, which must be at least two sites.

We summarize the context variables and related studies in Table 3.

TABLE III. CONTEXT VARIABLES AND RELATED STUDIES

Study	Context variable	Research topic	Specific variable	Year	Related study
[14]	Subject variables Task variables Site variables	Overlap differences	All (fictional maps and students)	2007	Unknown
[5]	Subject variables Task variables Site variables	Number of sites	All (fictional maps and students)	2010	[14]
[16]	Subject variables Task variables Site variables	Methodology	All (fictional maps and students)	2012	[5] [14]
[18]	Site variables	Feasibility	Number of sites	2005	[5]
[19]	Site variables	Process	Number of sites	2014	[5]

- Dependent variables

In general, studies performed in FTS investigate software development speed [5] [10] [13] [14] [16] [18]. It makes sense, because the FTS is about to reduce the overall development time. However, some studies are also looking by accuracy [5] [16] [17] and quality [16] [17] [19] as show Table 4.

Treinen and Miller-Frost [13] conducted case studies in FTS. Their studies were focused on the relation between speed and FTS approach. Inspired by this study, Espinosa, Nan and, Carmel [14] assess speed in terms of overlap between sites.

In 2009, Carrmel, Espinosa and, Dubinsky [15] conducted a field study focused in working speed. Inspired on this study, Solingen and Valkema [5] executed a controlled experiment to evaluate speed considering the number of sites in a FTS scenario.

Kroll et al. [16] also investigated speed. Their study was inspired in Carmel, Espinosa and, Dubinsky [15]. However, their study also focuses on other dependent variables such as, accuracy and quality.

Relate to quality, studies investigated this variable from two different points of view: methodology adopted [16] and handoffs management [19].

Hess and Audy [17] investigate the efficiency of a process to perform handoffs. Efficiency was evaluated in terms of accuracy and quality.

Kroll et al. [18] investigate the FTS's feasibility. Feasibility classified as an independent variable in FTS research. Thus, the feasibility was evaluated in term of speed outcomes.

Recently, Kroll et al. [19] investigated handoffs' management. Speed and quality are dependent variables present in this study. Previous studies [17] [20] provide best practices and recommendations to perform handoffs effectively.

In Table 4, we present dependent variables and related studies.

TABLE IV. DEPENDENT VARIABLES AND RELATED STUDIES

Study	Dependent variable	Research topic	Year	Related study
[13]	Speed	FTS	2006	[11]
[14]	Speed	Overlap between sites	2007	[11] [13]
[15]	Speed	Working speed	2009	[12] [14]
[5]	Speed Accuracy	Number of sites	2010	[14]
[16]	Speed Accuracy Quality	Methodology	2012	[5] [12] [15]
[17]	Accuracy Quality	Handoff	2012	[12] [20] [21]
[18]	Speed	FTS	2013	[11] [13] [16]
[19]	Speed Quality	Handoff	2014	[17] [20]

III. A FRAMEWORK FOR RESEARCH ON FOLLOW-THE-SUN

Based on a comprehensive review and analysis of the literature, we propose an initial research framework to support empirical studies and meta-analysis for research on FTS. The proposed framework is illustrated in Fig 1.

A. *Overview of the framework*

The proposed framework consists of three components: independent variables, context variables, and dependent variables.

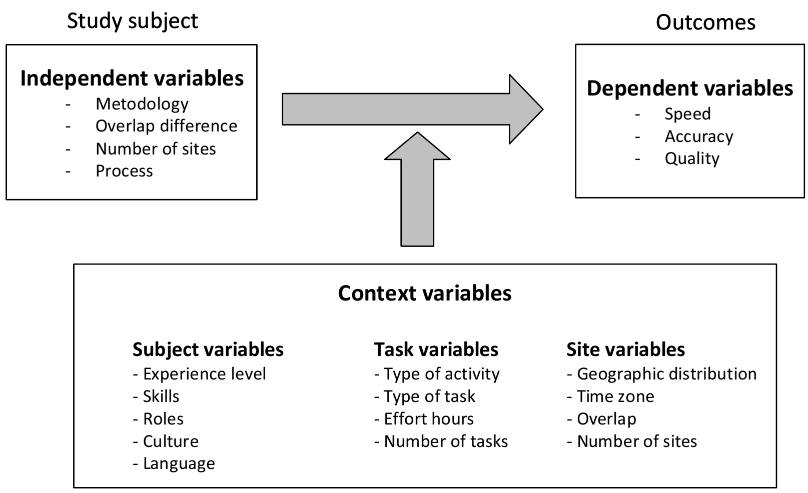

Fig. 1. The proposed initial framework for research on FTS

Research begins with the selection of least one independent variable. Independent variables are the study subjects. An independent variable is associated with the cause and is changed as a result of activities of the investigator and not of changes in any other variables [8]. Independent variables are analyzed under context variables selected to perform the study.

Context variables reflect the conditions under which the hypotheses regarding a relationship between the independent and dependent variables have been studied [9]. A context variable will always belong to one context [8].

A dependent variable is what is measured in the study and what is affected during the study. The dependent variable responds to the independent variable. It is called dependent because it "depends" on the independent variable. Dependent variables are outcomes of the study in terms whether or not the research project failed to deliver the expected result is given by one or more dependent variables defined in the study [8].

In order to obtain a better understanding of the relationship between different variables in the framework and key issues for research on FTS, we describe in the following sub-sections each variable considering its importance to further data analysis.

B. The Components of the Framework

1) Independent Variables

Empirical studies have investigated FTS using different project settings, processes and approaches. Also, empirical studies have made comparisons between methodologies, overlap differences between sites, number of sites FTS, different teams design and automated processes to implement FTS sub processes. These studies evaluate independent variables and its outcomes.

We identified five independent variables in empirical studies in FTS. In the next, we describe these variables.

- **Methodology:** two studies from the FTS literature discuss software development methodologies for FTS [12] [16]. In software engineering, there are adaptive (agile methods) and prescriptive (traditional methods) methodologies for software development. Methodologies are important because they can collaborate with FTS development. Some methodologies may be more promising for FTS. The methodology chosen for the project should be employed to achieve optimal productivity. In recent studies, agile methods such as, XP (Extreme Programming) and Scrum are the most discussed for FTS.
- **Overlap difference:** in FTS, team members work collaboratively in different time zones. Overlap differences between sites is a key aspect to perform handoffs in FTS development. Espinosa, Nan and, Carmel [14] observe the importance of overlap difference between site to increase the communication quality and to perform daily handoffs cycles. Additionally, overlap difference between sites has a strong relation with the team coordination and communication.
- **Number of sites:** GSD projects can have many development sites in the same project. In FTS, the number of sites suitable for FTS was investigated by Solingen and Valkema [5]. According to them, the importance of the number of sites for a FTS project is realized by the increasing on productivity. Productivity (usually measured by number of finished tasks) is frequently considered to determine the speed. The number of sites is a variable also present in context variables.

- **Process:** a process is used to designate a set of tools, methods, and practices used to produce a software product [22]. A process definition has a great influence on productivity. Thus, processes have been investigated in the FTS area to identify issues and also identify potential solutions and improvements to the overall process. Software processes are evaluated in terms of benefits and stronger tools and practices.
- **Feasibility:** two studies discuss how feasible is FTS approach for GSD projects [11] [18]. The first study reports the project failure. However, the second study claims FTS as feasible. The variable feasibility is used to evaluate and analyze the potential of a proposed approach, process, model or software practice which is based on extensive investigation and research. Furthermore, the feasibility variable is focused on gaining an understanding of the present applicability of the expected results.

2) Context Variables

We identified thirteen context variables. These variables are categorized by subject, task and site variables. We present the context variables as follow:

- **Subject variables:** empirical studies in FTS consider human factors to select participants. We call these factors as subject variables. The subject variables include in the framework are experience level, skills, role, culture and language. These variables are very important predictor of the success of a software project.
- **Task variables:** task characteristics can affect the outcomes of the FTS. Empirical studies describe the task variable as type of activity, type of task, effort hours and number of tasks.
- **Site variables:** geographic distribution, time zones, overlaps and number of sites are important variables that can reflect on the teams 'performance. A project manager considers these variables to plan the project, to define the software development methodology. Moreover, sites variables drive FTS.

3) Dependent Variables

A dependent variable can be viewed as the outcome of the study. We identified four dependent variables in empirical studies in FTS. These studies adopt qualitative and quantitative data analysis. FTS exiting studies reported many effects, measurements and outcomes in software development. The dependent variables in FTS research are described below.

- **Speed:** is the main dependent variable for researching in FTS. Due to team's distribution in different time zones, speed has a direct relationship with teams' efficient, increasing the efficiency of the project calendar and reducing the software development life cycle time (time-to-market). Speed is measured by the number of effort hours to complete a task, activity or software project.
- **Accuracy:** accuracy is considered as the degree of produced work represents the expected work, comparing by a previous specification [5]. FTS teams have to pass a task in progress to another team in order to continue the working day. It could impact on the working accuracy. Working accuracy is quantitative measures, measured by analyzing the produced work.
- **Quality:** is an important factor for successful projects [23]. Quality has a strong relationship with accuracy. Quality is usually measured by division of working accuracy by amount of delivered work. However, FTS is not about quality, but speed [1]. The relationship between quality and speed is observed in studies in order to identify FTS benefits.

4) Discussion

The proposed initial framework reflects on the state-of-the-art of FTS research. We observed a few studies with professionals and studies performed in the software industry. According to Carmel, Espinosa and, Dubinsky [4], the lack of FTS experience in the software industry is a barrier for FTS adoption. Additionally, case studies in FTS does not describe specific research details such as, number of subjects or tasks, demonstrating inconsistencies in some studies.

Most of the controlled experiments performed in FTS did not use software development tasks. It because software tasks require are more complex and also it require more time. Thus, it is difficult to compare results between studies.

As the FTS main goal is to increase software development speed, the most studies discuss this variable. Other dependent variables like accuracy and quality are also discussed in some studies. However, FTS does not aim to increase accuracy or quality. According to Carmel and Espinosa [4], FTS does not offer other advantages beside speed.

Relate to dependent variables, we observe the using of some metrics. These metrics are different between studies, but they aim to measure the same variables. For instance the variable speed was measured by Kroll et al. [16] and Solingen and Valkema [5].

The number of participants (context variable) is also diversified between the existing studies. Espinosa, Nan and, Carmel [14] conducted an experiment with 84 students. On the other hand, Solingen and Valkema [5] and Kroll et al. [16] conducted empirical studies with small teams.

The difference between existing studies does not guarantee that variables selected are more important or unique. Thus, we consider the selection of variables as a decision made by the researcher considering study's characteristics.

C. Directions for Future Research

The existing research on FTS present the need of solutions to move beyond the theoretical to the practical perspective. We observe the importance of studies clarifying how results from the literature promote solutions to implement FTS effectively.

Empirical studies in FTS have been adopted software development scenarios almost perfect. However, in practice, there are many constraints in global software projects. Thus, future research should consider exploring risks and survivability models for FTS.

FTS is an immature research topic [4]. We observed the lack of studies to define methods, practices and processes for FTS. Once, context variables are strictly defined, it is possible to investigate the effects on new methods, practices and processes for FTS.

Studies in FTS discuss mainly speed. Few studies discuss accuracy and quality. However, these variables are not less relevant for FTS. Low quality and accuracy can affect negatively a software project.

We observed FTS as a powerful strategy for global software projects. However, few software companies are able to realize the full benefit of FTS [3]. Each company decides how to implement a FTS project based on its technologies, availability of team members and sites. Thus, future research should explore how to define context variables to minimize the impact on the results achieved by dependent variables.

In this paper, we summary variables investigated by empirical studies. The proposed framework provides an overview of the research on FTS. New variables from future research can be added to the framework. Also, we observed the need of studies to define metrics for dependent variables. Finally, experienced researchers as new researchers in FTS should consider the challenges and the relationship between the existing variables to plan and conduct new studies.

Researching in FTS has many challenges. Studies report these challenges as some study limitations or obstacles. We highlight the main research challenges reported by studies:

- **Technical environment:** the lack of tools and technologies to support FTS development is reported by many studies [12] [17] [18]. Thus, empirical studies performed in FTS adopt tools and technologies developed to support GSD projects. FTS as a subset of GSD can take benefit of the solutions proposed for GSD. However, the technical environment of GSD does not support all FTS characteristics.
- **Technical background:** usually participants of FTS studies do not have experience with FTS projects. Prikladnicki and Carmel [2] report the Brazilian case, where managers involved in a FTS project did not understand or take advantage of the speed potential from a strategic perspective. Thus, FTS researchers have been investing in trainings before the study start. In some cases, participants also do not have enough experience in software development. According to Carmel, Espinosa and, Dubinsky [4] different technical backgrounds are usually adopted in GSD projects. However, it can impact in expect the results of FTS. Technical background as an obstacle for research in FTS discussed by Yap [12], Treinen and Miller-Frost [13] and, Kroll et al. [18].
- **Introducing process changes:** FTS differ from the GSD in that the development iterations tend to be very compressed due to constantly changing circumstances driven by real-time changes in tasks [1]. Thus, introducing process changes in software development scenarios may impact on the team's performance [12] [18]. For example, FTS couldn't be made to work by a large Brazilian company in 2010. FTS attempt failed and the project was terminated [2].
- **Lack of sponsorship:** empirical studies conducted in software companies' environments require financial investments from the company case. Companies have to allocate human resources and provide infrastructure to develop a study. Although, FTS is receiving considerable interest from companies over the world, still difficult to combine theory and practice. According to Endres and Rombach [24], developing theories without sufficient evidence regard to their usefulness in practice do not contribute to solve problems.
- **Lack of cost estimates for the project:** cost estimates for FTS projects did not take into consideration the time required for the overall coordination of the local and remote development environment [13]. Thus, in situations where rework is required due to an incorrect understanding of the requirements often has an effect on the overall project schedule.

IV. CONCLUSIONS

Based on a review of the literature, this paper synthesizes research variables in the existing FTS literature and presents an initial framework to support future research in FTS software development. The framework can be adopted as a first step in order to plan new empirical studies and to develop new theories about FTS. We followed a rigorous process to build the framework and categorize the variables. At least two researchers discussed each variable and its components. The categorization of the variables in independent, dependent and context variables provides an understanding of the FTS research. Researchers can make decisions about study subjects, study planning and expected outcomes based on the proposed framework. The proposed framework represents more than 20 years of the research in FTS. Since 1999, empirical studies performed in FTS are present in the framework represented by its variables.

Although FTS is an important research topic of GSD that has attracted the attention of researchers and software companies, researching FTS still a challenge. Thus, this study will help to focus research efforts on areas of practical importance in FTS and thereby help advance the state-of-practice in this area. In this study, we also hope to motivate future research in FTS, increase the number of studies and solutions for a successful FTS implementation.

Future research is needed to explore underlying theoretical issues in FTS. For example, researchers should consider the GSD context to gain a deeper understanding of FTS and to extend the research findings to build software process, methods and best practices for FTS. Moreover, researchers should also consider combining studies to gain a more complete understanding of how FTS can achieve its goal. In the future studies, we also aim to investigate how variables impact the collaboration, communication and culture.

ACKNOWLEDGMENT

We thank the PDTI Program, financed by Dell Computers of Brazil Ltd (Law 8.248/91), and CNPq (309000/2012-2). This research is also partially funded by the Rio Grande do Sul State funding agency (FAPERGS), projects 11/2022-3 and 002062-2551/13.

REFERENCES

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

[2] R. Prikladnicki and E. Carmel, "Is time-zone proximity an advantage for software development? The case of the Brazilian IT industry". Proceedings of the 2013 International Conference on Software Engineering (ICSE '13). IEEE Press, Piscataway, NJ, USA, 2013, pp. 973-981.

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

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

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

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

[8] C. Wholin, "Experimentation In Software Engineering: An Introduction", International Series in Software Engineering, Kluwer Print, 2000.

[9] H. Gallis, E. Arisholm and, T. Dyba, "An initial framework for research on pair programming," *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*, vol., no., pp.132,142, 30 Sept.-1 Oct. 2003.

[10] J. Kroll, S. I. Hashmi, I. Richardson, and J. L. N. Audy, "A Systematic Literature Review of Best Practices and Challenges in Follow-the-Sun Software Development", In Global Software Engineering Workshops (ICGSEW), pp. 18-23, 2013.

[11] E. Carmel. "Global Software Teams: collaborating across borders and time zones". Published by Prentice Hall-PTR, 1999.

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

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

[14] J. A. Espinosa, N. Nan and E. Carmel, "Do Gradations of Time Zone Separation Make a Difference in Performance? A First Laboratory Study". International Conference on Global Software Engineering, Munich, Germany, August, 2007, pp. 27-30.

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

[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.

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

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

[19] J. Kroll, I. Richardson, J. L. Audy, and J. Fernandez, J., "Handoffs Management in Follow-the-Sun Software Projects: A Case Study", In 47th Hawaii International Conference on System Sciences (HICSS), Hawai'i Island, USA, 2014.

[20] N. Denny, S. Mani, R. Sheshu, M. Swaminathan and J. Samdal, "Hybrid offshoring: Composite personae and evolving collaboration technologies," Information Resources Manage- ment Journal, 21, 1 (January–March 2008), 89–104.

[21] 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.

[22] S. L. Pfleeger, Software Engineering: Theory and Practice – 2. Ed. – Prentice Hall, 2004.

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

[24] A. Endres and D. A. Rombach, "Handbook of Software and Systems Engineering: Empirical Observations, Law and Theories". Addison Wesley, 2003.