Singapore Management University
Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

5-2011

Configuring global software teams: A multi-company analysis of productivity, quality, and profits

Narayanasamy RAMASUBBU
Singapore Management University, nramasub@smu.edu.sg

Marcelo CATALDO
Carnegie Mellon University

Rajesh Krishna BALAN
Singapore Management University, rajesh@smu.edu.sg

James HERBSLEB
Carnegie Mellon University

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Software Engineering Commons

Citation

Configuring Global Software Teams: A Multi-Company Analysis of Project Productivity, Quality, and Profits

Narayan Ramasubbu
Singapore Management University
80 Stamford Road
Singapore 178902
+65-68280669
nramasub@smu.edu.sg

Marcelo Cataldo
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
+1-(412)-268-2491
mcataldo@cs.cmu.edu

Rajesh Krishna Balan
Singapore Management University
80 Stamford Road
Singapore 178902
+65-68280879
rajesh@smu.edu.sg

James D. Herbsleb
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
+1-(412)-268-8933
herbsleb@cmu.edu

ABSTRACT

In this paper, we examined the impact of project-level configurational choices of globally distributed software teams on project productivity, quality, and profits. Our analysis used data from 362 projects of four different firms. These projects spanned a wide range of programming languages, application domain, process choices, and development sites spread over 15 countries and 5 continents. Our analysis revealed fundamental tradeoffs in choosing configurational choices that are optimized for productivity, quality, and/or profits. In particular, achieving higher levels of productivity and quality require diametrically opposed configurational choices. In addition, creating imbalances in the expertise and personnel distribution of project teams significantly helps increase profit margins. However, a profit-oriented imbalance could also significantly affect productivity and/or quality outcomes. Analyzing these complex tradeoffs, we provide actionable managerial insights that can help software firms and their clients choose configurations that achieve desired project outcomes in globally distributed software development.

Categories and Subject Descriptors

D.2.9 [**Management**]: productivity, programming teams, software process models, software quality assurance

General Terms

Economics, Management

Keywords

Globally distributed software development, software engineering economics, quality management, empirical analysis

1. INTRODUCTION

Global software development has become a dominant operational model for developing and delivering software-intensive systems. Software development organizations continue to expand and disperse geographically in search of lower project cycle-times, the ability to deliver high quality software at a lower cost, and the right talent necessary to develop the increasingly more complex software systems. However, realizing the benefits of dispersing software development does come with associated costs and challenges. Over the last decade, the negative impact of geographic distribution in software development has been well documented (e.g., [4, 7, 13, 16, 22]). Physical separation often introduces barriers that hinder software engineers' ability to adequately communicate and coordinate their work as well as stay aware of other peoples' activities [20]. Problems in communication, coordination, and awareness [9] lead to longer times to complete development tasks [7, 16] as well as more errors [12], which ultimately result in more software defects and development costs.

Given the rapid growth of the global software development model in spite of the documented theoretical and empirical evidence of the pitfalls and challenges of the model, it is important to understand how global software teams are still able to reap benefits from distributed development. Recent research has highlighted ways in which some teams have mitigated the challenges of global distributed development through appropriate process investments and quality management practices [2, 22]. This study takes a step further by examining the relationship between the *configurational* characteristics of the software project teams and project performance.

Configurational characteristics of distributed software teams refers to the structural properties of the geographic dispersion and accounts for the different ways in which the developers are dispersed, in addition to the well-studied time zone and distance measures (i.e., the spatio-temporal characteristics). Theoretical support for the need to study configurational characteristics comes from recent organizational studies that have established geographic dispersion as a multi-dimensional construct with each dimension having distinct implications for the psychological, social, and organizational processes that underlie individuals' ability and willingness to adequately communicate, coordinate, and collaborate [9, 18, 19].

There is also an emerging body of empirical evidence supporting the call for a more careful consideration of how the nature of work distribution impacts critical outcomes in software development projects. For example, a very consistent result reported by researchers is that development tasks take longer in a distributed setting versus a co-located one (e.g., [7, 16]). However, a software project represents a collection of development tasks and the loss in productivity at the task-level could be reduced or overcome by increasing the concurrency of project activities. In the context of software quality, time zone separation reduces the opportunities for synchronous interaction leading to higher levels of misunderstandings and errors [12]. "Nearshoring" (locating teams nearer to the customer in terms of time zones) has been proposed as an alternative that alleviates these detrimental effects [5]. Recent work has also shown that the nature of dependencies and

the ways in which developers are dispersed across the development locations is significantly more detrimental to quality than the impact of time zone separation per se [8]. These examples illustrate the need to develop a better understanding of how the different dimensions of geographic dispersion *jointly* impact performance outcomes such as development productivity and software quality.

In this paper, we examine the impact of several configurational dimensions of geographic dispersion on the productivity, quality, and profit outcomes of distributed software development projects. We collected data from 362 projects from four different companies. These projects spanned a wide range of programming languages, application domain, process choices, and development sites spread over 15 countries and 5 continents.

Our analyses revealed that project configurations that achieve high productivity tend to achieve low quality and vice versa, suggesting that managers face a challenging optimizing task when configuring their project teams. Further, imbalances in the configurations are often beneficial to project profitability, presenting a complex tradeoff between the three performance outcomes of productivity, quality, and profits. The key contributions of this paper are the following:

1) We take the first step in rigorously establishing the relationship between configurational dimensions of dispersion with multiple project performance outcomes that are considered jointly. This departs from prior work that typically considers each performance outcome in isolation, and therefore missing the interrelated tradeoffs between different performance outcomes and project-level configurations.

2) We analyzed the complex tradeoffs induced by project configurational choices and provide actionable insights that can help development managers and clients choose configurations that achieve desired outcomes in a distributed software development environment.

3) We collected and analyzed a large dataset spanning multiple companies, programming languages, processes, application domains, and geographical locations, which significantly improves the generalizability of our results.

2. GEOGRAPHIC DISPERSION AND SOFTWARE PROJECT PERFORMANCE

As global software development projects become pervasive, the nature of the distribution of development work and its relationship to outcomes of distributed projects have gained increasing attention from software engineering researchers (e.g., [2, 8, 14, 16, 21, 22]). This line of research has typically considered the spatio-temporal characteristics of work distribution, distinguishing between co-located and distributed development using measures such as the number of locations involved or the time zone separation [7, 12]. A recent stream of research in organizational studies has started to examine the multi-dimensional nature of distribution by considering configurational characteristics of distributed teams in addition to spatial and temporal dispersion measures [9, 18, 19].

Spatial dispersion refers to physical separation among the engineers participating in a development project, and researchers have typically measured it as the distance between team members. The key implication of spatial dispersion is the reduction in impromptu communication, which leads to lower levels of awareness of what the distributed project members are doing, the decisions that they are making, or the problems that they might have [18, 20]. Such a barrier to communication and awareness could result in coordination breakdowns and software integration problems [14, 23], which in turn could lead to longer time to complete development tasks, increase in rework, and/or higher numbers of defects [8, 16].

Temporal dispersion refers to the time zone differences among project members, and its key impact on distributed work is quite distinct from spatial dispersion. Temporal dispersion reduces the possibilities of synchronous interaction, which is a critical communicational attribute for real-time problem solving and design activities [20]. Managing the flow of information in asynchronous interactions can be quite complex. For this reason, temporal dispersion makes misunderstandings and errors significantly more likely to happen [6, 12]. On the other hand, temporal dispersion may allow distributed groups to accelerate the completion of development tasks using approaches such as "follow the sun", i.e., 24 hour round-the-clock development [24].

Configurational dispersion refers to structural properties of the geographic dispersion of teams and projects, and it can be disaggregated into three elements: the number of sites involved in a project, the evenness of distribution of project members across those sites, and the work experience level variations of project members at each site. Increasing the number of sites in a project could increase the complexity of the coordination effort. The coordination challenges could stem from more site-spanning dependencies as well as other factors such as diverse national cultures, regulatory boundaries and processes [4, 20]. On the other hand, if the development effort can be partitioned into autonomous units of work without any complex dependencies, increasing the number of locations helps concurrent development and might not per se translate into lower project performance.

The second aspect of configurational dispersion, evenness of distribution of team members across locations, influences the dynamics of interaction and coordination among development locations. For example, an uneven team distribution could promote behaviors such as majority influence and conflict, which are not conducive for good collaboration [18]. In an imbalanced configuration, members of larger locations might disregard the input from members of smaller locations and attempt to impose "majority" decisions that are inconsistent with the requirements of other locations, negatively impacting project outcomes. While such problems can be averted if project members are distributed evenly across different locations, an even team distribution might not make a good business case, especially in the presence of significant labor cost differences across the different locations.

Finally, the third aspect of configurational dispersion is experience dispersion, which has not received much attention in the literature. However, anecdotal evidence from our field observations suggests that it is an important factor affecting performance. Experience dispersion refers to how individuals with particular skills or levels of experience are distributed within the various development locations of a project. A concentrated presence of domain experts in a particular location could be beneficial to achieve good results as experienced engineers tend to produce better software faster [3, 10]. On the other hand, such a concentration of experts in one location could be associated with large numbers of less experienced individuals in other locations,

which might increase the likelihood of integration-related problems and the resulting loss of productivity and quality.

The discussion of the different dimensions of distribution presented in the previous paragraphs highlights two important gaps in the literature. First, the impact of the different configurational dimensions of work distribution might be varied depending on the unit of analysis. The direction of a particular effect at the level of the development task can be different or non-existent when considering the project as the unit of analysis. Second, we still lack systematic empirical evidence to understand how the various configurational dimensions of distributed teams jointly impact project performance. This leads to the two interrelated research questions examined in this paper:

RQ1: How do the configurational dimensions of distribution impact the performance (productivity, quality, and profits) of globally distributed software projects?

RQ2: What are the team configurations that firms can use to mitigate the ill effects of spatio-temporal dispersion, and overcome the challenges of distributed software development?

3. RESEARCH METHODOLOGY

In order to answer the research questions described above, we approached several firms that have adopted globally distributed software development models in their operations. The data corresponding to 362 projects, tracked from start to finish, from four firms were used in our study. The data we collected were audited by the quality assurance and central process engineering groups of the firms, and hence we can place high confidence in the reliability of the archival data. We augmented the archival data from the projects with structured interviews and field observations of the processes followed at the firms. Similar to prior research [2, 8, 22] we used a series of econometric models to examine the empirical relationship between the work dispersion measures and software project performance.

3.1 Description of Datasets

We describe the projects and firms in the four data sets in the following sections.

3.1.1 Dataset 1

The first firm in our dataset is a development organization that specializes in custom software development. It is privately held and headquartered in the USA with development teams located in Europe and India. The firm was certified for ISO 9001:2000 quality processes, followed the RUP development process model, and had an established clientele in USA, Europe, and Japan. We collected data on 45 projects executed by the firm between January 2007 and December 2009. The projects involved developing payment and billing enterprise solutions for the telecommunications industry using the J2EE, .NET, and PHP platforms. All the projects were executed using fixed price contracts between the firm and its clients.

3.1.2 Dataset 2

The second company we collected data from produces business-to-business solutions for electronic transactions. We collected data from 15 projects that were completed between 1999 and 2001. This company had 5 development locations distributed across Mexico, Ecuador, Brazil, and Argentina, and utilized agile processes for its development. All development sites followed a subset of agile processes with daily builds and periodic short status meetings among the members of the project. All developers had access to the same version control system and a task tracking system. The organization also used an internal instant messaging tool for communication between developers along with standard teleconferencing. The projects were built using Java-based technologies (Java, J2EE etc.) and fixed price contracts.

3.1.3 Dataset 3

Our third data set is from a multi-national company that produces complex embedded systems for the automotive industry. The software organization used a product line approach for developing its solution offerings. In such a setting, a base version of all the architectural components of the software system is first developed and then integrated to form a platform on which specific solutions are further developed. This company had 9 development centers located in North America, Europe, India, and Japan. We collected data on 56 projects that were completed between 2005 and 2007. These projects involved analyzing requirements of a particular customer, determining if the platform needed to be modified, and implementing those necessary changes in collaboration with the engineers that developed the platform. While the India-based development center of the firm was successfully assessed as operating at CMM level 5, the projects spread across the different development centers followed a RUP process model with extensive tailoring at the individual development centers. This set of projects used time and material contracts and developed their solutions using a mix of C, C++, and Assembly languages.

3.1.4 Dataset 4

The fourth company we collected data from is a multi-national firm specializing in custom software development with a large client base and development centers in Australia, Western Europe, India, and USA. All the development centers of the firm were assessed at CMMI level 5. We were able to collect data on 246 projects completed by the firm during 2007 and 2008. The projects involved developing custom web-enabled business software in J2EE and .NET platforms for the financial services industry. The projects were executed using fixed price contracts with clients with provisions for performance-based adjustments.

3.2 Description of Measures

We describe the key measures in our data set in this section.

3.2.1 Dispersion Measures

As stated earlier, we collected data on the *spatial, temporal,* and *configurational* characteristics of the work dispersion that occurred in the observed projects [18]. The data regarding development locations and number of developers per location was extracted from the project documentation (modification requests reports and change logs), process databases, and human resource database at the firms. For each pair of development locations we observed in our project, we determined the distance between these two development sites (using Google Maps and other map tools) to calculate the *Spatial Distribution* measure. We determined the difference in time zones between the development locations in order to calculate the *Temporal Distribution* measure. The third dimension of distribution, *Configurational Dispersion,* which is a key focus of this study, was assessed using three measures: Number of Sites, Personnel Imbalance, and Experience Spread. The descriptions, formulas, and examples of how we calculated the dispersion measures are shown in Table 1.

Table 1. Dispersion Measures

Dispersion Measure	Description	Formula	Example of calculation
		i-j: a location pair, K– no of location pairs, n_i – team size at location i, n_j – team size at location j, N – total team size in the project	Team size 10, distributed across 3 locations (location 1=New York; 2=Frankfurt; 3=Bangalore) with 5 people located in New York, 3 in Frankfurt, and 2 in Bangalore.
Separation *(spatial)*	Geographic distance among team members	$$\frac{\sum_{i-j}^{k} Miles_{i-j} * n_i * n_j}{(N^2 - N)/2}$$	Separation Difference = ((3858*5*3) + (8316*5*2)+ (4607*3*2)) / ((100-10)/2) = 3748.27
Time Zone *(temporal)*	Time difference among team members	$$\frac{\sum_{i-j}^{k} Time\ Zone_{i-j} * n_i * n_j}{(N^2 - N)/2}$$	Time Zone Difference = ((5*5*3)+(9*5*2)+(3*3*2))/((100-10)/2) = 4.07
Number of Sites *(configurational)*	Number of locations where team members work	K = Total number of development sites used in the project	Number of Sites = 3
Personnel Imbalance *(configurational)*	Extent of unevenness in distribution of personnel across locations	[Standard Deviation (n_i, n_j,…,n_k)] / N	Imbalance Difference = [Standard deviation (5,3,2)] / 10 = 0.15
Experience Spread *(configurational)*	Extent of unevenness in work experience of personnel across locations	[Standard Deviation (Average Team Experience at location i, Average Team Experience at location j,…,Average Team Experience at location k)] / N Note: The experience value for an individual is defined as the number of years that the individual has worked in professional software development.	Assuming that the average team experience at New York = 10, Frankfurt = 15, Bangalore =5, Experience Difference = [Standard Deviation (10,15,5)]/10 = 0.5

3.2.2 Project Performance Measures

We utilized three project performance measures – productivity, quality, and profits.

Productivity: We measured productivity using the following ratio of output and input parameters, where *Code Size* is measured using KLOC (and Function Points), and *Total Project Effort* is measured using person hours:

$$Productivity = \frac{Code\ Size}{Total\ Project\ Effort}$$

Quality: We measured quality as the delivered code defect density as following, where *Defects Delivered* were the number of defects reported by the client after the project had been delivered:

$$Quality = \frac{Code\ Size}{Defects\ Delivered}$$

Project Profits: We measured project profits using the following measure:

$$Profit = \frac{Project\ Revenues - Project\ Costs}{Project\ Costs}$$

We intentionally used ratio measures for all the project performance measures listed above as this allows us to control for the project size, scope, and scale effects across our four companies (the ratios normalize the code size, costs, and personnel effort across the different projects in our data set). This allows us to then rigorously compare measures across the four companies with a high degree of confidence.

3.2.3 Control Measures

Other than the dispersion and project performance measures, we also collected information on project team size and process models followed in the projects. These control measures are needed to ensure proper statistical model specification and analysis as these measures are known to have a significant effect on project performance from prior research, but are not the main focus of our analysis. Team size and project processes are well established as having impact on project performance [15, 17]. These additional variables, along with the 4 firm-level dummy identification variables are used as control measures in our empirical models.

Team size: The total full time equivalent count of personnel involved in a project at all development locations is our team size variable.

Process Model: Through structured interviews with team leaders and team members, and through our field observations of the projects, we gathered information on the process models followed in the projects. We encountered two development process approaches in the projects in our data sets: projects that followed a highly plan-based waterfall or V-model approach for development and projects that followed an agile, cyclical or spiral approach to development. In our dataset, there was a complete correlation between the waterfall development approach and the adoption of CMM process templates, whereas the projects that followed a cyclical or spiral approach to development had hybrid process templates based on both agile RUP and CMM. To control for these differences in process approaches across the projects we created dummy (categorical) variables in our dataset (1= structured, waterfall or V-model approach, 0=agile, iterative

process approach), and used these categorical variables in our statistical models.

Firm-Level Controls: Finally, we clustered the projects according to their firm-level characteristics, such as the contract choice and the programming language used. This allowed us to robustly compare results across projects with different firm-level characteristics.

3.3 Description of Empirical Models

In order to examine the associations between the dispersion measures and project performance measures, we developed and tested our empirical models. The functional forms of the empirical models were derived from prior research [1, 15, 17, 22] utilizing the economic production function view of software development (i.e., project performance outcomes expressed as a production function of personnel-related, process-related, and project-related input variables). Corresponding to the three performance outcome measures we studied, the three equations that form our empirical models are shown in Equations 1 to 3. It is important to note that our modeling approach analyzes the associations between the dispersion measures and performance outcomes by jointly accounting for the individual effects of various dimensions of dispersion, and the performance measures are estimated simultaneously accounting for any potential interrelationships between them.

In our dataset the spatial (Separation Dispersion) and temporal (Time Zone) dispersion measures were highly correlated with each other (pair-wise correlation of 0.9 with P-value=0.00). This indicates that our data involves East-West geographical dispersion, which correlates highly with the Time Zone classifications of the world, rather than North-South geographical dispersion. Hence we used only the separation dispersion measures in our empirical models to jointly stand for spatio-temporal dispersion (and omitted the time zone measure).

ln (development productivity) $= \alpha_0 + \alpha_1$* Separation Dispersion + α_2* Number of Sites + α_3* Personnel Imbalance + α_4* Experience Spread + α_5* Quality + α_6* ln(Team Size) + α_7* Process Model + ε_1 (clustered by dataset)... **(Eq. 1)**

ln (quality) $= \beta_0 + \beta_1$* Separation Dispersion + β_2* Number of Sites + β_3* Personnel Imbalance + β_4* Experience Spread + β_5* Internal Defect Density + β_6* ln(Team Size) + β_7* Process model + ε_2 (clustered by dataset) ...**(Eq. 2)**

Profit $= \gamma_0 + \gamma_1$* ln(Development Productivity) + γ_2* ln(Quality)+ γ_3* Number of Sites + γ_4* Personnel Imbalance + γ_5* Experience Spread + γ_6* ln(Team Size) +ε_3 (clustered by dataset)...**(Eq. 3)**

The summary statistics of the variables in our equations are shown in Table 2. We log-transformed two of the dependent variables (development productivity and quality) to ensure normality of the variable distributions. We estimated Equations 1 to 3 using Seemingly Unrelated Regression method. This approach accounts for the correlation of error terms in the equations (i.e., we assume that any external and firm-level events such as technology boom

and bust cycles, CEO resignation, etc. would impact all the three performance outcomes simultaneously) leading to a more accurate analysis. We also clustered the projects as per the datasets to derive robust standard errors and corresponding P-values. Since our models posit interrelationships between the performance outcomes, there is a potential for the independent variables to be correlated with error terms of the regression models, an issue known as the endogeneity problem. The implication of such a problem is the potential for biased estimates resulting in inaccurate inference. Therefore, we tested for presence of endogeneity using the Durbin-Wu-Haussmann test [11], and found no evidence of it. We also conducted other regular regression diagnostics such as testing for outlier sensitivity (using Cook's Distance), multicollinearity effects (using Variance Inflation Factors), and verified that our results are robust.

Table 2. Summary Statistics (N=362)

Variable	Unit	Mean	Std. Dev.	Min	Max
Productivity	KLOC / Person Hour	0.01	0.01	0.00	0.05
Quality	KLOC / Defect Count	10.06	23.29	0.01	240.05
Separation Dispersion	Miles	341.76	750.33	0.00	5035.50
Number of Sites	Count	3.00	0.66	2.00	6.00
Personnel Imbalance	Std.dev (Experience) / Team Size count	0.38	0.23	0.00	0.71
Experience Spread	Std.dev (Experience) / Team Size count	0.37	0.33	0.00	1.87
Team Size	Count	11.26	12.08	2.00	116.00
Process Model	Categorical	0.86	0.35	0.00	1.00
Code Size	KLOC	34.81	53.09	1.00	595.76

4. RESULTS: PROJECT PERFORMANCE

The results of our regression estimation are presented in Table 3. Our empirical models are statistically sound, (i.e., they are statistically significant and they pass all endogeneity, multicollinearity and other diagnostic tests) and have good explanatory power to explain the associations between the dispersion measures and project performance outcomes (Chi-Squared values are highly significant and adjusted R-Squared values are similar or higher than typical values documented in prior research).

Overall, our results indicate that variations in the configurational characteristics of distributed teams lead to different performance outcomes. For example, from Table 3, we notice that an increase in the number of sites and in the imbalance in personnel configurations both boost productivity at the expense of quality (the regression value for both these variables is positive (0.44 and 0.76 respectively) and significant for the productivity column and negative (-0.83 and -0.97 respectively) for the quality column). On the other hand, an increased spread in experience across the sites is associated with improved quality but decreased productivity. We discuss these results in more detail, for each of the three performance measures, in the next sections.

4.1 Work Dispersion and Productivity

Our results indicate that as firms distribute their software development across longer distance (and time zones) they benefit from improved project level productivity. Specifically we notice that a unit increase in separation dispersion measure (i.e., one standard deviation of separation dispersion measure ~750 miles) leads to 1.16 KLOC per person hour increase in productivity (refer to coefficient 3 in Table 3, $e^{0.15}=1.16$). A unit increase in the number of sites boosts project productivity by 1.55 KLOC per person hour (refer to coefficient 4 in Table 3, $e^{0.44}=1.55$). Similarly, increasing personnel imbalance across the sites by one standard deviation increases productivity by 2.14 KLOC per person hour (refer to coefficient 5 in Table 3, $e^{0.76}=2.14$). That is, our results show that, holding all else constant, a ten person project team configured as 8-2 between New York City and Bangalore is more productive than 6-4 and 5-5 configurations. It must be noted that these increases in productivity follow an exponential pattern and taper off at higher levels of the dispersion measures (i.e., a slower marginal increase or smaller slope).

Any imbalance in experiences of personnel significantly decreases productivity. A standard deviation increase in personnel experience difference between development sites causes productivity to decrease by about 0.4 KLOC per person hour (refer to coefficient 6 in Table 3, $e^{-0.89}=0.41$). That is, a project team configured across New York City and Bangalore with personnel having equal average work experience (e.g., 6-6 years) is more productive than a project team configured with an imbalanced experience profile (e.g., 10-2 years) even though the average project team experience in both configurations would be 6 years. These effects of the dispersion measures on productivity are visually shown in Figure 1. The graphs showing marginal effects were plotted using the regression coefficients reported in Table 3, holding all other variables at their mean levels.

Table 3. Regression Results (N=362)

Variable		ln (Productivity)	ln (Quality)	Profits
Productivity	1	NA	NA	**2.49 (0.09)** †
Quality	2	**0.34 (0.00)** *	NA	**1.79 (0.00)** *
Separation Dispersion	3	**0.15 (0.03)** *	**-0.24 (0.06)** †	**2.36 (0.09)** †
Number of Sites	4	**0.44 (0.00)** *	**-0.83 (0.00)** *	**-4.98 (0.03)** *
Personnel Imbalance	5	**0.76 (0.01)** *	**-0.97 (0.05)** *	**9.92 (0.03)** *
Experience Spread	6	**-0.89 (0.00)** *	**0.63 (0.03)** *	3.13 (0.16)
ln(Team Size)	7	**-0.95 (0.00)** *	**-0.31 (0.08)** †	**6.25 (0.05)** *
Process Model	8	**-2.4 (0.00)** *	**3.24 (0.00)** *	NA
ln(Code Size)	9	NA	**1.09 (0.00)** *	**-4.38 (0.00)** *
ln(Internal Defect Density)	10	NA	**-0.22 (0.00)** *	NA
Chi-Squared (P-value)	=	**233.17 (0.00)** *	**131.27 (0.00)** *	**4.13 (0.00)** *
R-Squared	=	0.45	0.35	0.15

Note: P-values in parenthesis; * statistically significant at <5% level, † significant at <10% level, results not in bold are not statistically significant. Firm-level dummy variables are not reported to preserve anonymity of firms.
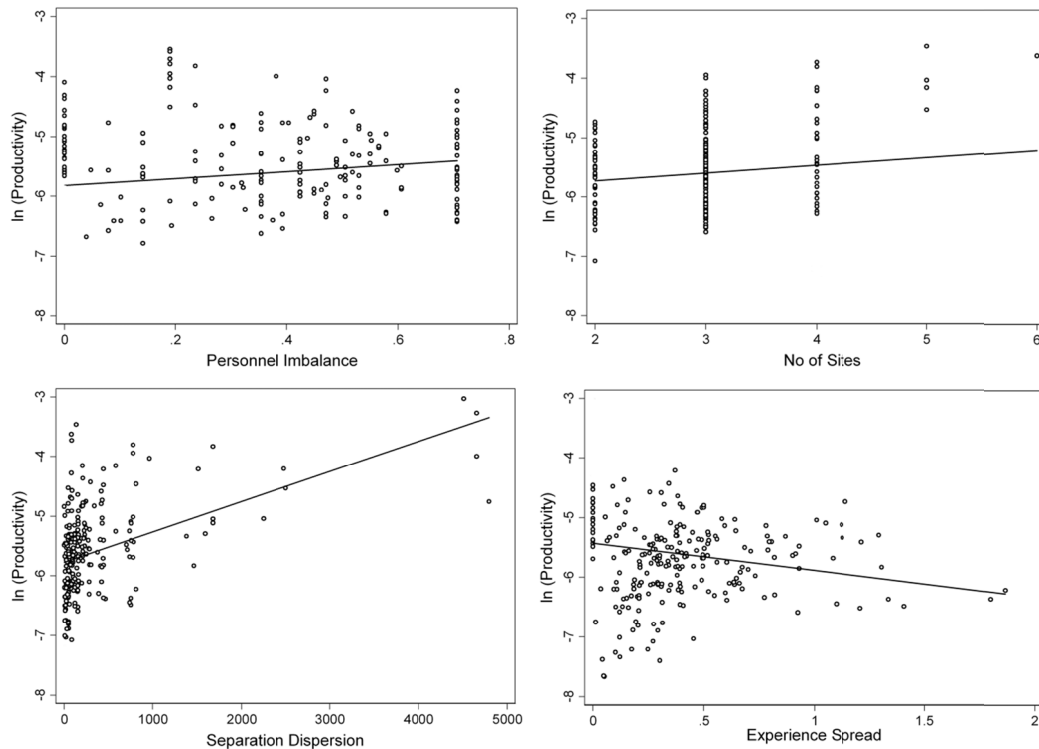

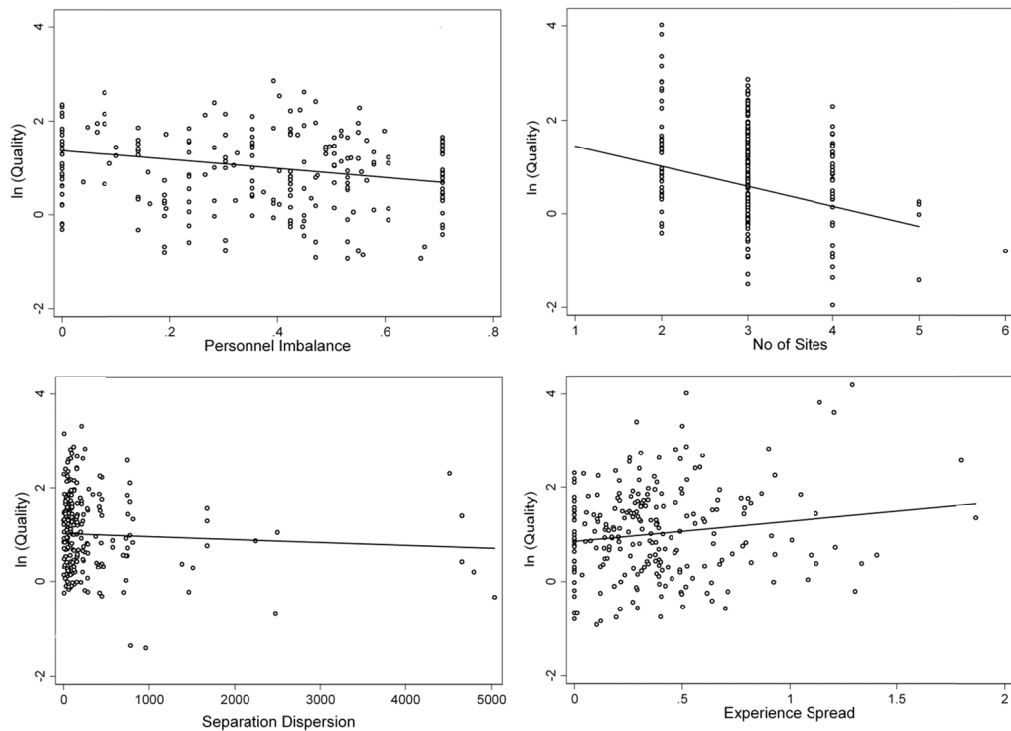**Figure 1. Effect of Dispersion Configuration on Project Productivity**

Figure 2. Effect of Dispersion Configuration on Project Quality

4.2 Work Dispersion and Quality

The effects of work dispersion measures on quality are opposite to their effects on productivity. As reported in Table 3, our results indicate that as firms distribute their software development across long distance and far away time zones, quality significantly decreases. Specifically, in our data set we notice that an increase in one unit of separation dispersion measure (i.e., a standard deviation increase in separation dispersion ~750 miles) leads to an increase in 1.26 delivered defects per KLOC (refer to coefficient 3 in Table 3, $e^{-0.24}=0.79$, $1/0.79=1.26$). A unit increase in the number of sites increases defects by 2.29 counts per KLOC (refer to coefficient 4 in Table 3, $e^{-0.83}=0.44$, $1/0.44=2.29$). Similarly, increasing personnel imbalance across the sites by one standard deviation increases defects per KLOC by 2.64 KLOC per person hour (refer to coefficient 5 in Table 3, $e^{-0.97}=0.38$, $1/0.38=2.64$). That is, our results show that, holding all else constant, a ten-person project team configured as 8-2 between New York City and Bangalore is more defect prone than 6-4 and 5-5 configurations. On the other hand, a standard deviation increase of imbalance in experiences of personnel across the development sites decreases the number of defects by 0.53 defects per KLOC (refer to coefficient 6 in Table 3, $e^{0.63}=1.88$, $1/1.88=0.53$). This indicates that a project team with a 10-2 experience spread across two locations produces higher quality software than a project team that has an experience spread of 6-6 (even though the average project experience in both configurations would be 6 years).

It must be noted that the increase or decrease in quality follow an exponential pattern and taper off at higher levels of the dispersion measures (i.e., a slower marginal increase or smaller slope). The effects of the dispersion measures on quality are visually shown in Figure 2. The graphs were plotted using the regression coefficients reported in Table 3, holding all other variables at their mean levels.

4.3 Work Dispersion and Project Profits

From our results reported in Table 3, we notice that higher productivity and quality positively contribute to better profit margins. Specifically in our data set we see that an improvement of 1 KLOC per person hour improvement in productivity leads to 2.49% increase in project profit margin (refer to coefficient 1 Table 3), and reducing one delivered defect per KLOC would increase profit margin by 1.79%. Increasing the number of sites of development by one count decreases profit margin significantly by 4.98% (refer to coefficient 4 Table 3). On the other hand increasing personnel imbalance by a standard deviation helps boost profits by 9.92% (refer to coefficient 5 Table 3), and dispersing development centers farther away in miles and time zones boosts profits by 2.36% (refer to coefficient 3 Table 3). Similar to personnel imbalance, an increase in experience spread is positively associated with improved profit margins (a weak statistical significance of single-tailed 10% level). These effects of dispersion configurations on project profit margin are visually shown in Figure 3.

4.4 Effect of Process Choice

Coefficient 8 in Table 3 shows that locally tailored, agile, and iteration-oriented software process models are associated with improved productivity outcomes whereas highly structured and plan-based models are associated with improved quality outcomes. Specifically, we see that agile methods are 0.09 KLOC ($e^{-2.4}=0.09$) per person hour more productive than structured methods while structured methods yield 0.04 less defects per KLOC ($e^{3.24}=25.53$, $1/25.53=0.04$) than agile methods.
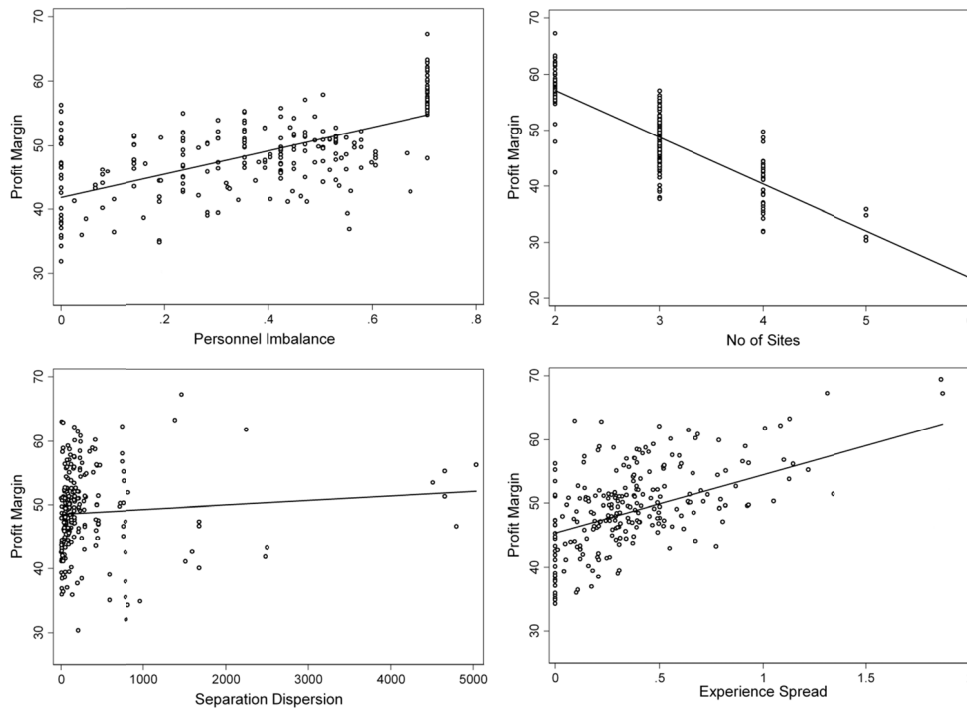
Figure 3. Effect of Dispersion Configuration on Project Profits

4.5 Results Summary

Table 4 summarizes the overall joint impacts of the configurational characteristics of software dispersion on key project performance outcomes. The results highlight a challenging tradeoff in configuring distributed development teams: while increasing both productivity and quality simultaneously is the most desired project performance goal, different dispersion configurations aid productivity and quality in diametrically opposite ways and disproportionately, making the simultaneous improvement in both productivity and quality using the same dispersion configurations an extremely challenging management issue. We discuss the potential reasons for our observed results, and the possible mechanisms that could help software managers manage the tradeoff in the next section.

Table 4. Empirical Results Summary

Variables	Productivity	Quality	Profits
Separation Dispersion	↑	↓	↑
Number of Sites	↑	↓	↓
Personnel Imbalance	↑	↓	↑
Experience Spread	↓	↑	↑

Note: Bolder arrows represent statistically significant results at two-tailed P-values <5% level and the arrow not in bold shows a result in the indicated direction, but not statistically significant at P-value < 5%.

5. RESULTS: CONFIGURATION OPTIONS

In this section, we address our second research question posed in section 2, and show the effects of different configurational choices on the three project performance outcomes. The set of possible team configurations is quite large in practice. To make these configurational choices easier to understand and given our available data, we focus on the experience spread and personnel imbalance dispersion measures only. In addition, we categorized the configurational choices in to three categories: 1) "Customer Site-Oriented Imbalance" (biased towards the customer site), 2) "Offshore-Oriented Imbalance" (biased towards the offshore development site), and 3) "Balanced" (fairly equally balanced between the customer and offshore sites). This allowed us to transform the large configurational choice space into intuitive categories that are still useful. We used only 294 projects for this analysis, as the remaining projects did not have the detailed data needed for this analysis.

Tables 5-7 show the results of our analysis for productivity, quality, and profit performance outcomes respectively. In each table, we rank order the effect of each configuration choice on the respective performance outcome. We have highlighted the top three choices in solid shades and highlighted the bottom two choices in stripes. The difference in mean-levels of outcomes (productivity, quality, and profits) between the top, middle, and bottom configurational choices, as tested by pair-wise t-tests are statistically significant at the 5% P-values.

The results show clearly the stark tradeoffs between productivity, quality, and profit. In particular, there are no configurational choices that are in the top three for all the three outcomes. In addition, the best configurations for profit are among the worst configurations for both quality and productivity – highlighting the challenge in selecting an optimal configuration. These results also reinforce and illuminate the earlier regression results. For example, the regression results (Table 3) show that productivity is positively associated with personnel imbalance and negatively

268

with experience spread. Table 5 supports this result by showing that a "balanced" configuration that balances the tension between personnel imbalance and experience spread has the highest productivity. Further, any imbalances should be towards the offshore-orientation to minimize the productivity loss. The results for quality (Table 6) show that the best quality is achieved when one measure (either experience spread or personnel imbalance) is balanced and the other is customer-oriented. This agrees and sheds more light onto our regression results, which show opposite effects for the two measures. Finally, Table 7 shows that the highest profits are achieved when both the personnel imbalance and experience spread measures are offshore-oriented. This agrees well with our regression results and our intuition, as the offshore sites tend to have lower labor costs.

6. DISCUSSION

6.1 Choosing a Dispersion Configuration

Our results indicate that choosing any specific configuration for a globally distributed software project induces fundamental tradeoffs between project productivity, quality, and profit. Fortunately, there appear to be some prescriptions that can be applied for rationalizing the tradeoffs and optimizing performance outcomes, which we discuss in this section.

6.1.1 Impact of Profit-Orientation

Our results show that by creating an imbalance of resources across the development centers (e.g., more headcount in "cheaper" locations) firms are able to improve their profit margins despite taking a dent in delivered quality (and the associated warranty and rework costs) and/or productivity (see Tables 5-7 and Table 3). This is because there is a significant cost arbitrage in software development labor costs. Our results show that distributed team configurations that depict larger spatio-temporal separation and personnel imbalances are going to be sustained simply because they are profitable to the firm. Hence, an important next step for researchers is to identify mechanisms that can help software teams manage the quality and productivity tradeoffs at the team or project-level when they use profit-oriented configurations.

6.1.2 Managing Tradeoffs

Given the inherent tradeoffs between productivity, quality, and profit, it is important that tradeoff-managing mechanisms are proposed that can help developers and/or clients choose configurations that achieve preferred outcomes. Our work provides this in two different ways. First, the prescriptions emanating from our regression results in Table 3 and the rank-ordering of configuration choices in Tables 5 to 7 can be used to choose configurational choices that are optimized for specific performance outcomes. Second, we observe that process choice can play an important role in determining performance outcomes. For example, teams that are originally configured to boost quality (e.g., lower spatio-temporal dispersion and personnel imbalances), could adopt a shorter planning range and more frequent release cycles as compared to the traditional waterfall model in order to improve their productivity. Similarly, teams that are originally configured for high productivity (e.g., larger spatio-temporal dispersion and experience spreads across locations) could boost quality and eventually profitability by incorporating structured processes and more disciplined planning.

Thus, we prescribe a dynamic and context-sensitive software development process environment that helps teams to alleviate the tradeoffs induced by specific project team configurations.

Table 5. Productivity

		Experience Spread Dispersion		
		Balanced	Customer Site-Oriented Imbalance	Offshore-Oriented Imbalance
Personnel Imbalance	Balanced	1	8	3
	Customer Site-Oriented Imbalance	6	5	4
	Offshore-Oriented Imbalance	2	No Data	7

Table 6. Quality

		Experience Spread Dispersion		
		Balanced	Customer Site-Oriented Imbalance	Offshore-Oriented Imbalance
Personnel Imbalance	Balanced	5	2	3
	Customer Site-Oriented Imbalance	1	4	6
	Offshore-Oriented Imbalance	8	No Data	7

Table 7. Profits

		Experience Spread Dispersion		
		Balanced	Customer Site-Oriented Imbalance	Offshore-Oriented Imbalance
Personnel Imbalance	Balanced	5	2	4
	Customer Site-Oriented Imbalance	8	7	3
	Offshore-Oriented Imbalance	6	No Data	1

6.2 Limitations of the Study

Our study has limitations worth highlighting. Our study only considered dependencies at the team level and not at the architectural or code level. Modular structures, for instance, tend

to exhibit a number of benefits in terms of quality and productivity. Therefore, an assessment of the technical coupling along with team level data could provide further insights. Second, despite the large number of projects from multiple companies, we were not able to examine the joint impact of spatial dispersion and temporal dispersion as these two measures were highly correlated in our dataset (due to our predominantly East-West site locations). Further research is required to examine the combined relative impact of both dimensions of distribution on development productivity and software quality. Finally, we have not looked at the individual-level leadership, social, and cultural factors that could possibly affect team dynamics and configurational choices that impact performance outcomes. We defer this investigation to future work.

6.3 Implications for Future Research

Our results suggest several areas for future investigation. First, it would be useful to understand the role that collaborative tools can play in reducing the tradeoffs in choosing configurational choices. For example, collaborative tools could facilitate easier coordination between locations, thus improving performance outcomes. We would also like to study the effect of complementary and conflicting process choices across multiple geographically distributed locations on various performance outcomes. Finally, we have anecdotal evidence through our field observations and interviews that project teams often face a need to overcome the tradeoffs in performance outcomes by dynamically changing their team configurations at different phases in the project lifecycle. We plan to investigate these issues in the future.

7. CONCLUSION

In this paper, we have analyzed the performance outcomes of 362 globally distributed projects from four different companies with respect to the configurational characteristics of those projects. We showed that there is a fundamental tradeoff between productivity, quality, and profits at the project configurational level. Finally, we provided actionable insights for development and client teams to choose configurations that help achieve preferred performance outcomes and mechanisms for shifting the observed tradeoffs.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] R. D. Banker and S. A. Slaughter, "The Moderating Effects of Structure on Volatility and Complexity in Software Enhancement," *Information Systems Research,* vol. 11, pp. 0219-0240, 2000.

[2] C. Bird, N. Nagappan, P. Devanbu, H. Gall, and B. Murphy, "Does Distributed Development Affect Software Quality? An Empirical Case Study of Windows Vista," *ICSE* 2009.

[3] W. F. Boh, S. A. Slaughter, and J. A. Espinosa, "Learning from Experience in Software Development: A Multilevel Analysis," *Management Science,* vol. 53, pp. 1315-1331, 2007.

[4] E. Carmel, *Global Software Teams: Collaboration across Borders and Time Zones*: Prentice-Hall, 1999.

[5] E. Carmel and P. Abbott, "Why 'Nearshore' Means That Distance Matters," *CACM,* vol. 50, pp. 40-46, 2007.

[6] M. Cataldo, M. Bass, J. D. Herbsleb, and L. Bass, "On Coordination Mechanisms in Global Software Development," *ICGSE,* Munich, Germany, 2007.

[7] M. Cataldo, J. D. Herbsleb, and K. M. Carley, "Socio-Technical Congruence," *Empirical Software Engineering and Measurement symposium*, Kaiserslautern, Germany, 2008.

[8] M. Cataldo and S. Nambiar, "On the Relationship between Process Maturity and Geographic Distribution: An Empirical Analysis of Their Impact on Software Quality," *FSE* 2009.

[9] J. N. Cummings, J. A. Espinosa, and C. K. Pickering, "Crossing Spatial and Temporal Boundaries in Globally Distributed Projects," *Information Systems Research,* vol. 20, pp. 420-439, 2009.

[10] B. Curtis, H. Krasner, and N. Iscoe, "A Field Study of the Software Design Process for Large Systems," *Communications of ACM,* vol. 31, pp. 1268-1287, 1988.

[11] R. Davidson and J. G. Mackinnon, *Estimation and Inference in Econometrics*. New York: Oxford University Press, 1993.

[12] J. A. Espinosa, N. Nan, and E. Carmel, "Do Gradations of Time Zone Separation Make a Difference in Performance? A First Laboratory Study," *ICGSE*, Munich, Germany, 2007.

[13] J. A. Espinosa, S. A. Slaughter, R. E. Kraut, and J. D. Herbsleb, "Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development," *Organization Science,* vol. 18, pp. 613-630, 2007.

[14] R. E. Grinter, J. D. Herbsleb, and D. E. Perry, "The Geography of Coordination: Dealing with Distance in R&D Work," in *International ACM SIGGROUP conference on Supporting Group Work*, Phoenix, Arizona, 1999.

[15] D. E. Harter, M. S. Krishnan, and S. A. Slaughter, "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," *Management Science,* vol. 46, pp. 0451-0466, 2000.

[16] J. D. Herbsleb and A. Mockus, "An Empirical Study of Speed and Communication in Globally Distributed Software Development," *IEEE Transactions on Software Engineering,* vol. 29, pp. 481-494, 2003.

[17] M. S. Krishnan, C. H. Kriebel, S. Kekre, and T. Mukhopadhyay, "An Empirical Analysis of Productivity and Quality in Software Products," *Management Science,* vol. 46, pp. 0745-0759, 2000.

[18] M. B. O'Leary and J. N. Cummings, "The Spatial, Temporal, and Configurational Characteristics of Geographic Dispersion in Teams," *MIS Quarterly,* vol. 31, pp. 433-452, 2007.

[19] M. B. O'Leary and M. Mortensen, "Subgroups, Imbalance, and Isolates in Geographically Dispersed Teams," *Organization Science,* vol. 21, pp. 115-131, 2010.

[20] G. M. Olson and J. S. Olson, "Distance Matters," *Int. Journal of Human Computer Interaction,* vol. 15, pp. 139-178, 2000.

[21] R. Prikladnicki and J. Audy, "Distributed Software Development: Toward an Understanding of the Relationship between Project Team, Users and Customers," *International Conference on Enterprise Information Systems*, 2003.

[22] N. Ramasubbu and R. K. Balan, "Globally Distributed Software Development Project Performance: An Empirical Analysis," *FSE* , Dubrovnik, Croatia, 2007.

[23] C. R. B. d. Souza, D. Redmiles, L.-T. Cheng, D. Millen, and J. Patterson, "How a Good Software Practice Thwarts Collaboration: The Multiple Roles of Apis in Software Development," *ACM SIGSOFT Software Engineering Notes,* vol. 29, pp. 221-230, 2004.

[24] J. J. Treinen and S. L. Miller-Frost, "Following the Sun: Case Studies in Global Software Development," *IBM Systems Journal,* vol. 45, pp. 773-783, 2006.