The Use of Information Systems by Collocated and Distributed Teams:
Towards the 24-Hour Knowledge Factory

Amar Gupta
Thomas R. Brown Professor of Management and Technology
University of Arizona

Satwik Seshasai
International Business Machines & Massachusetts Institute of Technology

Copyright: Amar Gupta and Satwik Seshasai, 2008. All rights reserved.
The Use of Information Systems by Collocated and Distributed Teams:  
Towards the 24-Hour Knowledge Factory

Abstract

Recent academic and policy studies focus on offshoring as a cost-of-labor driven activity that has a direct impact on employment opportunities in the countries involved. This paper broadens this perspective by examining the 24-hour knowledge factory as a model of information systems offshoring that leverages other strategic factors beyond cost savings. A true 24-hour knowledge factory can ensure that progress is made on tasks at all times of day by utilizing three or more sets of information systems professionals, located at strategically selected locations around the globe. Some organizations today utilize a service provider framework in which an offshore site provides service to the central site. Entire tasks are often outsourced to the lower-cost overseas site and sent back when completed. In contrast, the 24-hour knowledge factory involves continuous and collaborative activities round-the-clock. By organizing tasks in this manner, the 24-hour knowledge factory offers the potential to reduce turnaround time to complete tasks, to reduce total costs, and to provide better overall performance. Previous studies have examined the performance of individual teams over time and have explored the benefits of distributing work to distant teams, but have not directly compared the effect of collocation versus geographic distribution in the context of the use of emerging information system technologies and methodologies. Since the concept of the 24-hour knowledge factory is still in the early stages of practice for semi-structured tasks, a controlled field experiment was conducted using two distributed centers in which the tasks were performed in a sequential manner. The use of information systems and the overall performance were studied using a collocated software development team and a distributed software development team. While the two teams used information systems in dissimilar ways at key points during a project, the overall performance of the two teams was largely similar at the macro level but differed at the micro level. In other words, one organizational structure is not inherently superior to the other, in terms of the ability to meet stated goals. While spatial and temporal separations introduce new challenges, these can be overcome with the deployment of appropriate collaboration technologies—and even leveraged for strategic advantage. In sum, our findings suggest that firms can apply the two-center and three-center work models to transition from a service provider framework in which offshoring is a short-term and unilateral cost-saving tactic to a strategic partnership between centers in which offshoring becomes a core component of the global corporate strategy.

Keywords: 24-hour knowledge factory; information systems; collocated teams; virtual distributed teams; offshoring; outsourcing; innovation; group process.
The Use of Information Systems by Collocated and Distributed Teams: Towards the 24-Hour Knowledge Factory

Amar Gupta
Thomas R. Brown Professor of Management and Technology
University of Arizona

Satwik Seshasai
International Business Machines & Massachusetts Institute of Technology

INTRODUCTION

The Association of Computing Machinery (ACM) released a report in 2006 that took a comprehensive look at offshoring, starting with a base assumption that offshoring is largely about cost savings (Asprey, Mayadas and Vardi 2006). The report described the opportunities for all nations to benefit based on economic theories of comparative advantage; however, it did not discuss the opportunity to utilize a globally distributed workforce to transform the dynamic environment within which offshoring is conducted. An editorial in MIS Quarterly suggested acceptance of the shift of information systems jobs to other countries and preparation of students for a global job marketplace (Weber 2004). Similar reports from the Institute of Electrical and Electronics Engineers (IEEE) and the National Society of Professional Engineers (NSPE) also described offshoring in the context of jobs being gained or lost due purely to cost savings between nations (IEEE 2004; White 2004). The NSPE report declared that offshoring should only be done when the talent cannot be found in the United States and suggested this policy direction as the correct means for preserving jobs within the United States. The IEEE’s approach was slightly less defensive, as they advocated training programs within the United States to help U.S. workers gain skills to compete with international workers. These reports all present the
employment aspect as a competitive zero-sum situation, where work can only be done in one
country or the other.

Earlier research in psychology and business strategy reinforced the idea that product
development could be done most productively in one location. Studies on inter-personal
communication have shown that geographic distance reduces the opportunity for face-to-face
interaction (Conrath 1973), which is considered necessary for transferring tacit knowledge
between individuals and organizations (Porter 1998; Sternberg 1991; Tallman et al. 2004).
Traditionally, physical distance is considered to be detrimental to inter-personal and inter-
organizational collaboration, leading many firms in the 1980’s and 1990’s to prefer collocating
large cross-functional teams at a single site (Eppinger and Chitkara 2006). However, recent
advances in information technology have enabled virtual distributed teams to work effectively
without meeting face-to-face (Cummings 2004; Mazneski and Chudoba 2000), suggesting the
potential for a new model of distributed knowledge production that can leverage geographic
distance for strategic advantage.

A number of relevant papers have been published by various authors in the Journal of
Strategic Information Systems. One paper emphasizes that in the emerging global economy,
policy strategies must be integrated with the technology and market strategies to survive in the
market (Jarvenpaa and Tiller 1999). In a second JSIS paper, 49 different information technology
configurations were studied over time to define a high level set of choices that the firms could
utilize to craft their respective IT outsourcing portfolios (Cullen, Seddon and Willcocks 2005). A
third JSIS paper analyzes the relationship between development, globalization, and IS research
(Walsham 2005). And a recent paper examined global software development from the viewpoint
of socialization and face-to-face meetings in the context of distributed strategic projects (Oshri,
Kotlarsky and Willcocks 2007). In addition, several special issues of JSIS have addressed a broad spectrum of issues related to IS, globalization, and distributed teams; this includes Volume 9, Number 2/3 that focused on knowledge management and knowledge management systems; and Volume 11, Number 3/4 that focused on trust in digital economy.

The view of offshoring as primarily a cost saving exercise has gradually transitioned to viewing offshoring as a mechanism for utilizing a globally distributed workforce in a new manner made possible by advances in information systems. While this may seem obvious to organizations practicing the model described in this paper, the literature in this domain lacks details of extended experiments, conducted in commercial environments, on how the use of tools and methods differs when increased handoffs exist, and how the distributed model can be made most effective. Transaction cost theory has been used as a model for studying IT outsourcing and helping to determine whether a global operation should be pursued. These studies use constructs such as loss of control over work (Loh and Venkatraman 1995), high transaction costs (Ang and Straub 1998), and threat of knowledge loss (Duncan 1998). These theories are based on an underlying model that suggests multiple sites should be treated as distinct entities. This paper focuses on a slightly different issue that encompasses three key facets: how can distributed teams work effectively with frequent transfer of work-in-progress with each other?; how can the individual teams work during daytime in their respective countries and still achieve round-the-clock operation?; and how effective can such geographically distributed teams be in comparison to collocated teams? These issues deserve attention because they will help to revise the traditional transaction cost model for deciding to pursue global work. As a motivating example, Levina and Vaast investigate how the differences in status between onshore and offshore participants in software development teams impact collaboration effectiveness [Levina and Vaast
Their research shows how status differences stem from limited access of offshore participants to important resources such as business knowledge, contacts with important decision makers, and control over spending decisions. Their research revealed that some middle-level IT managers on the client side were able to renegotiate status differences by sharing their resources with offshore teams, by promoting an atmosphere of respect towards offshore professionals, and by building joint practice and identification across sites.

By employing this methodology both within the development of IS tools to facilitate the model, as well as within organizations producing the information systems, it becomes possible to consider offshoring in a mutually beneficial perspective where the interests of workers in high-income economies are aligned with workers in other countries and customers worldwide. One example of an IS-facilitated paradigm for globally dispersed workforce is the 24-hour knowledge factory model (Gupta and Seshasai 2004), which advocates continuous work on knowledge-based tasks by individuals located in time zones that allow for 24-hour engagement. The role of task and team familiarity has been studied in other software development contexts by researchers, notably by Espinosa et al (Espinosa et al 2007). Each individual in the globally distributed work environment works the normal workday hours that pertain to his or her time zone, and then passes the task to a fellow worker located in a different time zone. An example of this phenomenon is seen in the automobile industry where the concept of platform shifting allows members of a product development team in one country to work on a prototype and to transfer the work-in-progress to another country at the end of the workday [PBS 2007].

The case study presented in this paper describes a more typical two site global work environment; however, the insights gained from the latter case study can serve as the basis for analyzing the characteristics of true 24-hour knowledge factories that are evolving in different industries over time. In particular, two kinds of environments are of direct relevance to the information systems community. One is the design, development, and implementation of information systems in a manner that leverages the distributed workforce paradigm. The other is the development of new information system approaches that will enable this paradigm to be
applied to a broad range of white-collar activities ranging from medical services to logistics planning, and from financial analysis to product design. The current thinking around tools related to distributed teams is that outsourcing presents a “challenge” that must be overcome with appropriate technology tools. This paper examines the benefits that may accrue from a team being distributed and altering the pattern of their use of such tools.

**Research Question and Foundations**

In order to formulate the core research question of this paper, a definition of knowledge factory is required. A knowledge factory is defined as a collection of knowledge-driven workers tasked with producing a knowledge-based asset, with the workers frequently creating incremental assets that are passed back and forth among fellow workers. A globally distributed call center is, in some ways, a knowledge factory because when calls are handled, the knowledge pertaining to the particular call is stored centrally and is available to the next individual who has to handle the same topic or the same caller. The software test and fix cycle environment is another knowledge factory in which software is the knowledge-based asset, and the knowledge of whether the software accomplishes its function is passed back and forth between software developers and testers. The 24-hour aspect, mentioned above, can be considered to be a manifestation of a knowledge factory where work is performed on a continuous basis around the clock.

*What are the implications of spatial and temporal separations between workers on overall productivity? What are the long term implications of the 24-Hour Knowledge Factory paradigm?* In order to answer these critical questions, one needs to look at historical precedents as well as to analyze a number of interrelated technical, strategic, organizational, and economic issues. The notion of shifts can be traced back to the industrial revolution.
manufacturing equipment was scarce and costly, different sets of employees were scheduled to work in successive shifts so that the manufacturing facilities could be used on a round-the-clock basis. The use of the 8-hour shift system evolved over time. Initially, each worker was directed to work 12-16 hours a day so that each machine could be used for an extended period of time. Then, the notion of having two shifts evolved. Based on new legislation on both sides of the Atlantic, the work hours were gradually reduced. The introduction of the shift system yielded benefits in terms of higher productivity of each machine, reduced production times, and lower prices to customers. However, it also created social and health issues by requiring the person to work in an urban setting, usually away from other members of the family, and also at odd hours and changing work schedules determined by the idiosyncrasies of the manager in charge of assigning workers to different shifts.

The advent of electronic computers and the diminishing costs for telecommunications enabled the development of 24-hour Call Centers. Depending on the time of the call, it is automatically directed to a call center that is active at that time. Using a cluster of three to four call centers located in time zones 6-8 hours apart from the time zone of the neighboring call center, one can ensure that all employees of these geographically distributed call centers are working during daytime in their respective countries. The notion of multiple support centers was subsequently adopted for supporting global communications networks over time.

By involving specialized microchip design engineers located at multiple places around the world, a semiconductor chip design firm creates virtual multi-site knowledge factories. This structure allows for an efficient design process with faster turnaround time, which is one of the major potential benefits of distributing work across time zones (Gupta and Seshasai 2004; Treinen and Miller-Frost 2006; see also Eppinger and Chitkara 2006; Majchrzak et al. 2000).
Suitable tasks should be modular in nature, so that natural breaks can serve as hand-off points. Personnel must also adapt, and be willing to trust and yield control to fellow team members who continue the same task in subsequent shifts.

Global workforces provide the firm with access to high-talent designers who would otherwise have to move to a different country, or work at odd hours of the night, often referred to as the “graveyard shift.” In the past, people around the world deemed the time difference between fellow workers to be a major impediment when implementing information systems. Now, the perception has switched around; for many projects, the time difference is viewed as a strategic plus. However, both views are based on largely untested assumptions regarding the nature of work by collocated and distributed teams and the feasibility of handing off tasks across shifts.

Research related to this area has been motivated partly by expert perspectives on the changing nature of organizations and IT. Venkatraman and Subramaniam (2002) have suggested that future research on strategy and competitive advantage should focus on individuals, intellectual capital, and relationships between individuals. The individuals involved are inevitably distributed around the globe and the challenge is to build systems that leverage global talent (Venkatraman 2004). The creative use of the available time afforded by a global workday is required to take full advantage of this opportunity. The treatment of temporal structures as defining characteristics of organizations can lead to further understanding of these organizations (Orlikowski and Yates 2002). Sambamurthy and colleagues (2003) tie these factors directly to the IT investments that firms make by providing a framework for understanding that IT investments are indeed key influencers of firm performance along with factors such as agility and capability-building. Others have shown that structural diversity in work groups, including
geographic diversity of team member locations, increases exposure to external knowledge sources and thereby enhances sharing of knowledge (Cummings 2004).

**Hypothesis**

This paper attempts to extend the guiding frameworks discussed above, delineate the core principals and corresponding research areas related to global work environments, and test the differences between collocated and distributed teams in a controlled field experiment. Our hypothesis is that information systems influence the differences in collocated team dynamics vis-à-vis distributed teams and that the creative use of information systems in managing global operations can affect the behavior of the team.

At a more detailed level, the authors felt that:

(i) The collocated team will demonstrate higher productivity, in terms of final outputs but without taking cost into consideration, because it would involve the overhead of transferring tasks back-and-forth on an incremental basis;

(ii) The distributed team will demonstrate lower costs because part of the staff is based in India and is paid lower salaries;

(iii) The distributed team will exhibit lower cost-benefit ratio if both productivity and costs are considered on a concurrent basis;

(iv) The distributed team will make greater use of automated collaboration technologies in order to overcome the inability for the two components of the distributed team to interact on a face-to-face basis; and

(v) The overall quality of the end-product will be roughly similar in the two cases.
While we were able to get access to detailed data on most aspects, details of costs were not available because of corporate policy and allied reasons. As such, the detailed hypotheses were reframed into the following set of 11 hypotheses, labeled as H1 thru H11 respectively.

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1:</td>
<td>The distributed team will rely more heavily on written communication for group discussion.</td>
</tr>
<tr>
<td>H2:</td>
<td>The distributed team will rely less (than the collocated team) on broadcast style email messages.</td>
</tr>
<tr>
<td>H3:</td>
<td>The distributed team will conduct longer discussions primarily in written (email) form.</td>
</tr>
<tr>
<td>H4:</td>
<td>The distributed team will send fewer logistical messages to members of the group.</td>
</tr>
<tr>
<td>H5:</td>
<td>The distributed team will make major use of the source code modification process to resolve issues, in place of informal collaboration, before the ‘feature freeze’ date.</td>
</tr>
<tr>
<td>H6:</td>
<td>The socio-technical system of the distributed team will be less interconnected (as compared to the collocated team).</td>
</tr>
<tr>
<td>H7:</td>
<td>The distributed team will rely more on meetings for short term issues.</td>
</tr>
<tr>
<td>H8:</td>
<td>The distributed team will formally assign tasks in meeting format.</td>
</tr>
<tr>
<td>H9:</td>
<td>The output of the distributed team will be similar, in terms of quality, as that of the collocated team.</td>
</tr>
<tr>
<td>H10:</td>
<td>The distributed team will rely more on formal systems for knowledge capture, as compared to the collocated team.</td>
</tr>
<tr>
<td>H11:</td>
<td>The productivity of the distributed team will be lower than that of the collocated team (because of the overhead involved in transferring tasks back and forth on an incremental basis).</td>
</tr>
</tbody>
</table>

The above hypotheses include a number of constructs that are further defined here. H1 through H4 describe written communication, which in this field study referred to electronic mail messages. H4 refers to logistical messages, which are defined as being related to a specific task or action to be completed in less than a week’s time. H5 refers to a source code modification process; this is a computer-based system for logging changes made to the computer programs being developed by the team. H5 also refers to a “feature freeze” date, a milestone used by both teams to complete programming tasks required for all features within the given software release. H6 refers to a socio-technical system, which in this case was tested by examining the discrete
modules of source code being developed (as distinguished by the teams), and the number of modules of source code that had more than one individual contributing to them. H7 and H8 refer to meetings that were conducted over the phone or in person. H9 refers to quality and H11 refers to productivity; both of these parameters were assessed through qualitative interviews.

The logic underlying these hypotheses is based on the notion that a distributed team requires more handoffs of knowledge, and thus requires more formal systems to facilitate these handoffs. Accordingly, the distributed team adapts the technical design and processes to reduce the number of interactions required. This has an impact on the nature of discussions, the nature of tasks, and the nature of assigning technical modules, as described in the hypotheses.

The above hypotheses were examined from the perspectives of previous researchers who have investigated issues of similar type; subsequently, a detailed case study was conducted at IBM, as described later in this paper.

**Group Dynamics and Tacit Knowledge**

There has long been a sense that face-to-face contact can facilitate creative interaction and produce more and better ideas (Osborn 1957). However, there is an equally long history of experimental findings that show that the aggregate output of so-called “nominal” or “concocted” groups of individuals working alone outstrips the aggregate output of “real” groups of the same number of individuals working together in person on creative tasks such as idea generation (Lorge et al. 1958; Mullen, Johnson, and Salas 1991; Taylor, Berry, and Block 1958). Real interactive groups consistently incur a “process loss” during group interaction that nominal groups avoid (Steiner 1972). Later experiments (Diehl and Stroebe 1987, 1991) found that the inability of all real group members to contribute their ideas simultaneously created a bottleneck that blocked potentially valuable contributions of some members and thereby reduced the
effectiveness of real groups. Subsequent studies have shown that providing real interactive
groups with information technology tools enabled simultaneous creative production and removed
social inhibition, thus eliminating the production blocking problem (Paulus et al. 1996). In fact,
real groups were sometimes found to be even more productive than nominal groups (Dennis and
Valacich 1993; Valacich, Dennis, and Connolly 1994). Since globally distributed teams share
key characteristics with nominal groups (members working more independently than
collaboratively), and also with electronic interacting groups (team member interaction mediated
by technology), the social psychology literature on small group dynamics implies that global
virtual teams may in fact enjoy certain advantages relative to collocated teams. With the aid of
electronic communication, the advantage of distributed teams over collocated interactive teams
grows even further as group size increases (Gallupe et al. 1992). Hypotheses H1 through H4
were formulated on the basis of these theories.

While electronic tools allow distributed teams to work interactively (to some extent) and
productively on creative tasks, they do not resolve the challenge of tacit knowledge, considered
essential to innovative activities but difficult to transfer without face-to-face interaction (Kogut
and Zander 1992; Nonaka, Toyama, and Konno 2000; Sternberg et al. 2000). The accessibility to
ambient tacit knowledge has been posited as a major reason for firms to locate in close
geographic proximity to other organizations within the same industry (Audretsch and Stephan
1996; Porter 1998; Rosenfeld 1997; Tallman et al. 2004). If correct, globally distributed teams
may be missing a key ingredient that would help them function more effectively, suggesting that
collocated product development teams may be preferable after all. Hypotheses H5-H8 and H10
were designed to examine the set of questions raised by these pieces of literature.
Previous studies that suggest contradictory advantages and disadvantages of distributed and collocated teams do not permit formulation of specific directional predictions about how each type of team would perform on a similar product development task.; H9 and H11 attempt to address this void. Furthermore, the literature on the advantages of geographic proximity in innovation and on group process and creativity is limited in several ways. For example, the case for the necessity of physical proximity to foster innovation is based mostly on selective anecdotal evidence, and has been refuted by a recent large-scale survey study of high-technology firms that found no impact of geographic distance on collaborative new product development outcomes (Ganesan, Malter and Rindfleisch 2005). Similarly, most experimental studies of group idea-generation and problem-solving examined only very small ad hoc groups (3 to 5 persons), composed of relative strangers (undergraduate students), who worked without leadership on a very short-term task (often only 10 to 60 minutes in duration) involving a problem of no inherent interest or relevance to the concerned participants (Dunnette 1964; Dunnette, Campbell and Jaastad 1963). In contrast, real-world product development teams consist of highly trained professionals working with similarly qualified colleagues on an engaging and urgent problem, often over an extended period of time ranging from many weeks to more than one year. Such teams are supervised by management, motivated by incentives, and their performance carries real consequences for individual members and their organization. Therefore, theoretical questions regarding the impact of collocation or distribution of members on the effectiveness of product development teams should be evaluated using real teams working in appropriate field settings and tested under controlled conditions.

Case Examples of Virtual Teams in a Global Context
A number of recent studies and examples of virtual teams were examined to provide context for our research. These studies were mostly anecdotal but offer initial evidence that global teams are capable of sharing knowledge and acting as local teams (Ishizaki 2005). Others have shown that roles can be effectively distributed across distant locations (Chandler 2001), and that fostering a sense of collective ownership can facilitate effective daily hand-offs of work items across shifts and around the globe (Yap 2005). Chakrabarty (2006) discusses real-life examples of outsourcing that mainly focus on the sourcing from one firm to another, and includes two relevant examples of the distinction between distributed teams (in this case, outsourced to a different organization) and collocated teams (in this case, insourced to the same organization). In both examples, Chakrabarty describes scenarios where the entire offshore team is managed by a project manager who serves as the single point of contact to the onshore client manager. This is true even in the case where the team is physically located alongside the client manager. The teams involved in the examples described by Chakrabarty deliver incremental features to the client manager whose team tests and provides feedback in a very formal manner. The study described in this paper goes beyond this example into the emerging practice of truly distributed teams where there is no notion of a single point of contact per location, or a distinction between onshore and offshore work.

FIELD STUDY

Method

In a recent longitudinal study of a virtual product development team distributed across three organizations and three distant locations, Majchrzak and colleagues (2000) collected ethnographic and quantitative data to develop an in-depth understanding of the operation of the team from a social, organizational and technical perspective. They monitored the team’s use of
technology and organizational processes in implementing the multi-site development effort
during the 10-month project using participant observation, recorded meeting minutes, electronic
log files, weekly questionnaires and periodic interviews. Data of this type allowed these authors
to analyze the team’s initial attempts to overcome misalignment through organizational changes,
followed by subsequent success after organizational changes were combined with technological
changes adapted to the team’s geographic distribution. Other researchers have examined multiple
virtual teams, ranging from multiple teams (Maznevski and Chudoba 2000) to a survey of
hundreds of teams (Cummings 2004); however, these studies sought teams with maximally
diverse structural characteristics and teams working on different types of projects.

Our longitudinal study of collocated and distributed software development teams at the
IBM Corporation follows the general approach of Majchrzak et al. (2000), in terms of collecting
similar comprehensive data, but with three key differences: (1) we compare two teams within
one firm (rather than one team across three firms); (2) we manipulate the key variable of
organizational structure (geographic distribution); and (3) we use more objective and finer-
grained data on team member interaction and project outcomes. In contrast to descriptive case
studies, the present study is a controlled field experiment that compares two teams with nearly
identical characteristics except for the critical variable of interest: collocation versus geographic
distribution of team members. The design is a “quasi-experiment” (Cook and Campbell 1979) in
the sense that team members were not randomly assigned to each type of team, but the twin
features of similar composition of team and exercise of controls for other possible explanatory
factors allows us to infer that the difference in the organization of the two teams was the basis for
observed differences in team performance. Each team was assigned the same number of
members (7) with similar qualifications and experience, and each team worked on a nearly
identical software development project (two parts of the same software package); they were subject to identical time schedule (12 months) and deadlines, in the same corporation, and under the same project management and work rules. Further, all the collocated team members worked on the software during the same work hours, whereas the globally distributed team members shifted work back and forth across time zones in an asynchronous manner. The two sequential work shifts of the distributed team are less than the three consecutive 8-hour work shifts in the ideal 24-hour knowledge factory model, but dispersion of the team across 10 time zones forced team members to work more independently during their respective shifts, providing a conservative test of the key feature of the model.

The field study employed state-of-the-art technology to collect more detailed and objective measures of group interaction and performance than previously published studies of product development teams. For example, software was written to extract vital data from source code control systems and bug report databases that track the specific knowledge work product in real time, instead of relying on retrospective and perceptual self-reports by individual team members. Quantitative and qualitative data were collected systematically from the two teams over a period of one full calendar year. Since the main project deliverable was on a one-year timeframe, this period covered every major point in the project lifecycle from its inception to the delivery of the end product. Within this year, the teams also devoted a significant amount of time to short-term tasks such as attending to customer deployment issues and fixing bugs for maintenance releases; as such, the one-year timeframe also provides an opportunity to gain insights on knowledge sharing for all scopes and varieties of tasks. Figure 1 depicts the inputs, processes, and outputs involved with the two IBM software development endeavors. The key differentiating input factor between the two groups was geography. All members of the
collocated team were based at a single location in the U.S. In the globally distributed team, some members were based at an IBM center in India and the rest at an IBM center in the U.S.

Figure 1: The Inputs, Processes, and Outputs related to the software teams studied in this paper. The team and product characteristics were the same, except for geography. These aspects feed into a set of four social processes that were studied in the context of the technical processes which they facilitate. The outputs shown were controlled as well – the two teams produced a relatively similar set of outputs.

Measures

During the 12 months of the software development project, several types of data were collected. The data sources and measures were designed to provide a complete picture of the knowledge sharing that occurred over time in terms of technical, organizational, social, and group process dimensions. The experimental design and quantitative measures enabled direct comparisons between the collocated and distributed groups on the key dimensions of interest.

Primary Data – Personal Interviews
Hour-long personal interviews were conducted with each of the developers on each team. While the focus of these interviews was primarily to gain qualitative insight, specific quantitative questions were asked in order to elicit the developers’ own views of their knowledge sharing requirements. Data on the following measures were collected:

- Number of informal interactions per week (informal interactions with fellow team members that did not begin with an intention of discussing business).
- Number of informal interactions with main developers.
- Number of formal interactions with main developers.
- Percent of informal communication in person.
- Percent of informal communication via instant messaging.
- Percent of informal communication via phone.
- Number of tactical decisions made informally (decisions that were minor in scope, i.e., with minimal knowledge sharing requirements and minimal impact on other developers’ work).
- Number of strategic decisions made informally (decisions that were major in scope, with significant knowledge sharing requirements and long-term impact on other developers’ work).
- Number of strategic decisions that were speeded up informally
- Number of tactical decisions that were speeded up informally

**Primary Data – Weekly Meetings over One Year, Coding System for Tasks and Status**

The weekly meetings of each team were analyzed to gain insights into the processes of formal task allocation and knowledge sharing, on a group-wide basis, for each team. The minutes were recorded by the project manager for the teams, who maintained item-by-item details of the discussion. The collocated team held one face-to-face team-wide meeting per week, while the distributed team held one weekly face-to-face meeting for only the U.S.-based team
members and one weekly coordination meeting via telephone between the development leads from the U.S. and India.

A manual review of the three meetings per week (one meeting for collocated team, one meeting for U.S. team members of the distributed U.S.-India team, and one U.S.-India team joint session) was conducted to distill the following data, with respect to each developer:

- Main Developers Interacted With.
- Tactical Tasks Assigned.
- Strategic Tasks Assigned.
- Tactical Status Requests.
- Strategic Status Requests.
- Developer Input Requested.
- Developer-to-Developer Information Requests.

Archival Data – Software Problem Reports

Each development team kept track of fixes requested or made to the code base via Software Problem Reports (SPRs). These SPRs contained information on the problem being reported, as well as the history of knowledge provided by various developers in resolving the issue and information regarding the actual fix to the issue. SPRs were stored in a central database for each team. For purposes of this study, a software tool was written to collect the data from the SPR archive. This tool analyzed the software problems that were fixed over the 12-month period of study, and collected the specific types of data described below, for each developer, on a weekly basis:
• Main Developers Overlapped With – For a given SPR’s primary developer, a listing of the other developers who provided input into the SPR.

• Average Delay Between Developer Inputs – For a given SPR, the average time between one developer’s input and another developer’s input.

• Ratio of Collaborative to Individual SPRs – A collaborative SPR is defined as one that includes input from more than one developer.

• Average Time to Resolution – The average time it takes for an SPR to move from being approved by the management team to being fixed, and finally to actually being logged as fixed.

Archival Data – Source Code Control System

Each of the two teams used a source control system to log the modifications made to each element of the source code for the team’s product. The source control system stored the date, time, developer making the change, and a comment regarding the particular change. The comments often cited particular SPRs if there was an SPR that initiated the particular change to be made.

The goal of collecting data from the source control system was to ensure a clear depiction of the technical system, which would complement the social and organizational systems described by the other forms of data that were collected. The latter data provided a representation of the technical dependencies between developers on the teams, and the rate of technical collaboration within the teams. The following data were collected, with respect to each developer, on a weekly basis:

• Main Developers Interacted With – in modules for which multiple developers check the code, and the rate of shared check-ins with each of the other developers.
• Delay Between Check-Ins – the average time difference between modifications to a particular module.
• Reciprocal Check-Ins – the rate of check-ins: one developer performs a check-in, followed in time by another developer performing a check-in to the same module.
• If SPR cited, Average Developer Input on SPR – if the comments in a source control check-in refer to an SPR, the SPR was consulted to determine the number of updates posted to the SPR.
• Average Time Since Last Check-In – this provided an idea of the amount of code that was actively being modified at that specific point in time.
• Average Modules Checked In Per Build – At periodic intervals, the source code was incorporated into an executable version of the product. The frequency of this activity ranges from once or twice daily in the testing and fixing stages of the project lifecycle, to once every week in the design and implementation stages of the project lifecycle.

Archival Data – Group Email Exchanges

A software tool was written to analyze electronic mail (e-mail) sent to all members of each team. A “thread” refers to the entire set of messages written in response to an initial electronic broadcast or request for information. These data provided insights into the use of broadcast messages to share knowledge on the teams. The following data were collected, with respect to each developer, on a weekly basis:

• Main Developers Interacted With.
• Threads Contributed To.
• Average Delay Between Responses For Initiated Threads.
• Number of Threads Initiated.
• Average Length of Initiated Threads.

• Average Number of Developers Per Thread Initiated.

## Results

The data collected above provides a dashboard of key indicators of the weekly functioning and performance of the distributed (offshore and onshore) and collocated (onshore only) teams over the entire 12 month period of the project. Comparisons of outcomes for the key process variables for the distributed and collocated teams, respectively, are presented in Table 1, based on the set of 11 hypotheses formulated earlier in this paper. The table contains means and standard deviations of each observed variable. Additionally, a test of statistical significance was applied to each dataset to validate the formulated hypotheses.

One of the major surprises was that members of the collocated team communicated more frequently via emails than did members of the globally distributed team; this is despite the fact that many of the collocated team members worked in the same hallway of the same building. On average, there were also more developers per code element for the collocated team, as compared to the distributed team.

The two teams differed most dramatically in the number of source code modifications prior to the “feature freeze” deadline in week 41, with the distributed team making 53.8 modifications compared to only 11.6 modifications by the collocated team. In other words, the collocated team was able to approach a key product development deadline with much fewer last-minute changes, and its work on the software code was more collaborative and involved more consultation than the work by the globally distributed team.
The two teams used team meetings for very different purposes; the distributed team meetings featured a significantly higher percentage of tactical (cf. strategic) agenda items and also much higher percentage of assignment items (cf. status items).

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Process Variable</th>
<th>Distributed Team</th>
<th>Collocated Team</th>
<th>T-test (p&lt;0.05)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1: The distributed team will rely more heavily on written communication for group discussion.</td>
<td>Contributors per email thread</td>
<td>1.73 1.55</td>
<td>1.50 0.74</td>
<td>Inconclusive</td>
</tr>
<tr>
<td>H2: The distributed team will rely less (than the collocated team) on broadcast style email messages.</td>
<td>Average weekly email threads</td>
<td>10.42 5.05</td>
<td>19.85 10.75</td>
<td>Confirmed</td>
</tr>
<tr>
<td>H3: The distributed team will conduct longer discussions primarily in written (email) form.</td>
<td>Average emails per thread</td>
<td>2.32 2.25</td>
<td>1.75 0.95</td>
<td>Inconclusive</td>
</tr>
<tr>
<td>H4: The distributed team will send fewer logistical messages to members of the group.</td>
<td>Average weekly emails</td>
<td>17.06 10.13</td>
<td>29.91 19.55</td>
<td>Confirmed</td>
</tr>
<tr>
<td>H5: The distributed team will make major use of the source code modification process to resolve issues, in place of informal</td>
<td>Source code check-ins prior to deadline</td>
<td>53.82 74.56</td>
<td>11.56 11.0</td>
<td>Confirmed</td>
</tr>
<tr>
<td>H6: The socio-technical system of the distributed team will be less interconnected (as compared to the collocated team).</td>
<td>Average number of developers per code element</td>
<td>1.10</td>
<td>0.2</td>
<td>1.63</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>H7: The distributed team will rely more on meetings for short term issues.</td>
<td>Fraction of tactical (vs. strategic) meeting items</td>
<td>0.81</td>
<td>0.17</td>
<td>0.39</td>
</tr>
<tr>
<td>H8: The distributed team will formally assign tasks in meeting format.</td>
<td>Percent of task assignment (versus status) meeting agenda items</td>
<td>0.35</td>
<td>0.13</td>
<td>0.24</td>
</tr>
<tr>
<td>H9: The output of the distributed team will be similar, in terms of quality, as that of the collocated team.</td>
<td>Average SPR actions per week</td>
<td>134.21</td>
<td>168.3</td>
<td>104.37</td>
</tr>
<tr>
<td>H10: The distributed team will rely more on formal systems for knowledge capture, as compared to the collocated team.</td>
<td>Average # of individuals modifying SPR state</td>
<td>3.25</td>
<td>0.97</td>
<td>1.74</td>
</tr>
<tr>
<td>H11: The productivity of the distributed team will be lower than the collocated team (because of the overhead involved in transferring tasks)</td>
<td>Average SPR time to resolution</td>
<td>113.80</td>
<td>83.17</td>
<td>120.72</td>
</tr>
</tbody>
</table>
Of the 11 hypotheses, 7 were validated by the data from the case study at IBM. The statistical tests for four others were inconclusive; these were H1, H3, H9, and H11. The last one is perhaps the most significant: we started with the premise that the productivity of the collocated team will be higher as it avoids the overhead involved in transferring tasks between team members on a frequent basis. This premise was not substantiated by the data based on standardized tests.

Despite the very different usage of information systems and meeting behavior, each team exhibited similar performance in terms of the quality and speed of their work (measured by weekly SPR actions and average time to resolve SPR’s, respectively). These findings are discussed in the following subsections.

1. Technologies and Processes Need To Be Applied Differently

As shown in Table 1, hypothesis 10, asserting that the distributed team relied more on formal knowledge capture systems, was statistically supported. In the case of minutes of meetings, while the agenda categories were generally the same between meetings, the number of tactical items and number of task assignments were much higher in the distributed team. Hypothesis 8, too, was confirmed by statistical analysis. The distributed team used the meeting process as the mechanism for addressing needs that the collocated team satisfied outside of the context of the formal weekly meeting.
Based on the above, we can assert that geographic distribution can cause a significant behavioral difference in the use of information systems. The two teams had very different ways of applying the same, or very similar, information technologies and processes. More individuals were involved in modifications to SPR’s in the distributed team suggesting that they used this technology as a means of transferring information between team members and maintaining a record of status, while the collocated team could rely on synchronous communication for purposes of information sharing and status reporting.

These differences suggest that technology and processes that support knowledge sharing can be designed to explicitly serve different purposes (cf. Maznevski and Chudoba 2000). Barley (1986) provides a framework for assessing the role of technology in a knowledge-based work environment and suggests that the context in which the work is performed can significantly impact the way the technology is used. Teams will gradually adapt available technologies to suit their specific spatial and temporal structures.

2. Social Relationships and Technical Behavior Are Linked

Hypothesis 6 was confirmed, highlighting that the collocated team had more examples of code elements that were modified by multiple team members; interviews confirmed that this was because of the greater degree of social interaction on this team, rather than any piece of software requiring more intertwined technical interaction than the other. The interview sessions also revealed many cases where casual interactions led to technical decisions. While such social relationships are much easier to form when the team is collocated, the experience of one U.S. developer on the distributed team who traveled to India suggests that social relationships can be built across distant geographic and cultural boundaries and these relationships can be leveraged to satisfy technical goals.
Based on the above, the degree of social interaction between developers on a team was shown to have an impact on the technical behavior of the team, which then led to tighter social relationships. Developers on both teams cited the comfort level between team members as being important in facilitating creative discussions, so that developers did not have to worry about feeling embarrassed by a poor idea.

3. Geographic Structure Does Not Define Output

The geographic structure of the teams in this study led to different forms of value being achieved from their knowledge sharing processes; however it does not follow that the output of each team is necessarily defined by its structure, as discussed below and as shown in Figure 2. The structure of the distributed team led its members to have a higher degree of documented decisions, as revealed by the data on the use of emails, tactical meeting items, and SPRs. The concerned hypotheses, 2 and 7, were confirmed to be significant, while the one on the use of SPR, hypothesis 9, was inconclusive due to large variability in the data. Interviews with members of the distributed team confirmed that a very valuable, though perhaps, unintended outcome of this documentation process was that the history of the decision making process of the team was better retained. The collocated team cited more frequent informal communications as a process that led to higher incidence of finding new and creative solutions. Even though these informal meetings generally occurred face-to-face, the distributed team could still achieve a similar outcome. Two suggestions, provided during interviews with the distributed team on this specific topic, were a one-time face-to-face meeting that would introduce team members and incorporate a social component to the relationships, and the use of explicitly informal phone calls where no agenda or topic was preplanned so that team members could discuss any open-ended
Based on the analysis of the data and the interviews, Table 2 summarizes the major advantages of globally distributed and collocated teams.

**Table 2**: Major advantages of globally distributed and collocated teams.

<table>
<thead>
<tr>
<th></th>
<th>Distributed Team</th>
<th>Collocated Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formal logging of knowledge</td>
<td></td>
<td>Issues resolved informally, in a timely manner</td>
</tr>
<tr>
<td>Structured use of processes</td>
<td></td>
<td>Incidental interaction led to productivity</td>
</tr>
<tr>
<td>Explicit role definition via tasks</td>
<td></td>
<td>Meetings focused on strategic discussion</td>
</tr>
<tr>
<td>Exploiting technology for collaboration</td>
<td></td>
<td>Technical system more collaborative</td>
</tr>
</tbody>
</table>

**Figure 2**: Example scenarios where the geographic structure of a team leads to a process that achieves a desirable outcome that can also be achieved in a different manner by the team whose structure does not naturally allow for the highlighted process. The alternative path to the desired outcome for each team is indicated by dashed lines and boxes. In sum, the tight link between team structure, group process and outcomes does not need to exist.
Table 2: Major Advantages of Each Type of Team

4. Advantages of Geographic Distribution

Our finding that both collocated and geographically distributed teams were capable of successful collaboration suggests that common themes in the literatures on offshoring (offshoring is a win-lose zero-sum proposition), innovation (geographic distribution is a barrier to overcome) and social and organizational psychology (face-to-face groups are less productive) may all be inaccurate. Numerous benefits from leveraging a dispersed geographic structure were cited in interviews with the distributed team. Examples include: an increase in documentation and history retention; enhanced ability to share short-term tasks which required immediate attention so that work could be performed around the clock; and a more structured and explicit definition of work tasks and distribution of work items.

Extension of Ideas to the 24-Hour Knowledge Factory Environment

As mentioned earlier, the concept of using three or more geographically dispersed sites is already in use by call centers. Over time, this notion has begun to be applied to applications of greater sophistication and with less inherent structure, placing greater reliance on information systems to provide the necessary collaboration. At the current stage, the goal is to address diverse applications that involve semi-structured work, both related to development of new information systems as well as to the development of information systems that can be used by other industries, as depicted in Figure 3.
The notion of 24-Hour operations using 3 or more globally distributed centers is being utilized in some call centers. Over time, this notion has begun to be applied to applications of greater sophistication, with less inherent structure, placing greater reliance on information systems to provide the necessary collaboration.

The three organizational scenarios depicted in Figure 4 illustrate the situations that may apply to a distributed software support center, a software maintenance engineering team, and a new product development environment respectively. In the autonomous scenario (first case), individuals work relatively independently and do not rely on others for advice in making their decisions. An example of this is a software support center where customers can call an individual support representative and receive knowledge from that one individual. In the semi-autonomous scenario (second case), individuals still work independently, but occasionally need to consult others with more expertise, creating a hierarchy. An example of this is a software maintenance engineering team assigned to develop incremental releases to an existing software product. Such a team can work somewhat independently because the changes to the code are primarily isolated bug fixes, but may occasionally need to consult experts such as the original
developers of the code. The third scenario involves individuals who are heavily interdependent, such as members of software new product development team; the decisions made by one team member have impact on many other team members, and also require the inputs of many other team members. For all these three scenarios, our research team has explored new techniques and concepts that could be applied in the context of the 24-Hour Knowledge Factory.

![Decision-making dependencies for individual work: three scenarios.](image)

**Figure 4:** Decision-making dependencies for individual work: three scenarios.

In the traditional paradigm, the “ownership” right to each component of software was typically vested in only one person, who was responsible for ensuring that the actual performance of the component matched its expected state, and for surmounting defects and inconsistencies. While multiple persons participate in the programming endeavor, the owner of the component bears ultimate responsibility for bug fixes and possesses the most comprehensive knowledge of that particular component. With sites that have zero overlap in work schedules, as in the case of the 24-hour knowledge factory, the single ownership model is no longer appropriate.

In order to mitigate development risks introduced by communicating across culture, languages, and time zones and to provide 24-hour access to code owners, the concept of composite persona (CP) was formulated (Denny et. al. 2008). A CP is envisaged as a highly
cohesive micro-team that, like a corporation, possesses simultaneous properties of both an individual and a collection of individuals. A composite persona is designed to act as a singular entity, even though it comprises of (at least) three individuals, with one person at each site, with one site being active at any point in time. This configuration does not imply, however, that all sites possess the same number of persons; this is because one developer may belong to more than one CP. Using this notion of CPs, development can proceed in a manner similar to the traditional one, with the sole difference that the owner of each component is not an individual, but a CP. This CP is actively involved in the process of development and conflict resolution on a round-the-clock basis.

The fundamental operation in the context of CPs is the handoff. The handoff procedure used during the implementation of the concept development prototype system is derived from the one used in the Scrum agile process (Schwaber and Beedle 2002). In the latter case, Scrum stand-up meetings occur for co-located developers, first thing in the morning. All attendees briefly summarize what they accomplished the previous workday, what problems they encountered, and what they expect to accomplish during the day ahead. In our case, such debriefing takes place via electronic means. Now, the notion of CP is being reinforced with a new software process, CPro (Sheshu and Denny 2007). CPro, inspired by the Personal Software Process (PSP) (Humphrey 1995), is an agile process that enhances the productivity of the CP by providing automated mechanisms to estimate delivery schedules, reduce defects, propagate knowledge within the CP, and heuristically assign workloads to members in a manner that will maximize overall productivity.

MultiMind is a new process-aware groupware tool that facilitates software development using the CP method of collaboration. MultiMind follows the CPro workflow but
also incorporates elements from the agile processes XP and Scrum. To accomplish this, MultiMind logs operations on project artifacts as well as knowledge events. Knowledge events are generated when a developer accesses some form of electronic memory. This memory may be the project knowledge-base, a previous e-mail, or even the World Wide Web. Project artifacts include program source code, code reviews, UML designs, schedule estimates, and others. By analyzing the Lifestream which holds the project knowledge-base, the Scrum handoff can be partially synthesized, reducing the load on the developers during handoff.

The MultiMind prototype is being augmented to cater to voice inputs. This will allow the programmers to annotate their code more easily and more liberally. The voice inputs will be translated into text form. One will therefore have the facility to click on a piece of code, and to know the rationale for it either in audio form or in text.

The prototype software is being evaluated by a team consisting of faculty and students in three countries: the Wroclaw University of Technology in Poland (WUT), the University of Arizona, Tucson, USA (UA) and the University of Technology, Sydney, Australia (UTS) (Chaczko, Klempous, Nikodem, & Rozenblit, 2006). Based on the initial results from the academic environment, further tests of the 24-Hour Knowledge Factory approach will be conducted in industrial settings.

**Conclusions**

In this paper, we explored the dissimilar usage of information systems by a collocated team and a distributed team, both engaged in the development of new software products at IBM. We found that offshore decentralization of information systems development can succeed with proper design and management of the dispersed team, and strategic use of information systems. At the outset, we referred to the 24-hour knowledge factory model as the evolving model for
leveraging geographic and temporal differences. We acknowledge here that many of the issues cited for the distributed team occur because there are handoffs between people involved. If the individuals were not in distributed locations, but still had the same number of handoffs between shifts, we suspect that many of the same results would have been observed. However, the distributed team is the primary instance where knowledge-based work will involve multiple handoffs and thus was chosen to be one of the key foundations of our controlled experiment.

The implications for information systems theory and practice of each of the key themes identified in our field study are summarized in Table 3. Our findings show that information systems can play a major role in facilitating work productivity and social relationships among globally distributed team members who do not have an opportunity to meet face-to-face during a project, supporting and extending prior findings on electronic group brainstorming (e.g., Gallupe et al. 1992). These systems facilitated effective group interaction while preserving some advantages enjoyed by “nominal” groups of individual team members working independently without process loss from direct personal interaction.

This study also demonstrated that the geographic structure of a team (collocated or globally distributed) does not predetermine team outcomes. With increased understanding of the dynamics of each type of team, information systems can be adapted to help each team increase its effectiveness. Neither structure is inherently superior; both are workable models with proper adaptations. The results also show that geographic distribution can be leveraged as an asset by taking advantage of the possibility of continuous engagement on tasks across time zones.

<table>
<thead>
<tr>
<th>Major Findings from Field Study</th>
<th>Implication for Information Systems</th>
</tr>
</thead>
</table>

35
Technologies and processes were applied differently, but yielded good results in both cases.

Information technologies and systems need to be adapted based on the structural characteristics and processes of each team.

Social relationships and individual productivity are intertwined.

Information systems must be designed to foster social communication among members of distributed teams.

Distributed and collocated teams behave differently at different stages of the project.

Information systems for distributed teams must be able to cope with spikes in communications between team members, especially just prior to milestones for globally distributed teams.

Geographic structure of team does not alone define productivity of the team.

The overall productivity of teams is defined by many factors, with the degree of sophisticated of the collaboration framework being one of them.

Geographic distribution can be an asset

Information systems that facilitate structured knowledge capture can exploit spatial distribution by leveraging continuous engagement on tasks.

<table>
<thead>
<tr>
<th>Table 3: Recommendations for Improving Team Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our case study amplifies and extends the findings of previous studies of virtual teams.</td>
</tr>
<tr>
<td>For example, Cummings (2004) focused on external knowledge sharing as opposed to intra-team knowledge sharing, and links structural diversity to a higher degree of external knowledge sharing. Our study focused primarily on internal knowledge sharing but reaches a similar conclusion that having structural diversity does lead to a change in knowledge sharing and knowledge reuse. We examined one specific form of structural diversity, which is the distribution of team members on a geographic basis. The information systems developed in this case study to capture data can also be used to capture external knowledge sharing of the type examined by Cummings, but at a more granular level.</td>
</tr>
</tbody>
</table>
The design used by Maznevski and Chudoba (2000) incorporated face-to-face visits; in our case, there was no face-to-face interaction at all. They found that communication patterns are adapted to the task, whereas in our case, the two teams worked on nearly identical tasks (a control in the experiment), yet the communication patterns that developed were drastically different. We conclude that the communication patterns are a result of structural diversity, and that one pattern is not inherently superior. Instead, we suggest that the patterns must be understood in order to realize the potential benefits of each type of structure.

Our study provides an extension to the research of Majchrzak and colleagues (2000) who concluded that the role of technology in facilitating virtual teams is more important than the organizational structure. We conducted a more granular analysis of the use of technology by mining the email system, source control system, and software problem report system of each team. We analyzed both the social and technical networks by looking at critical factors such as the number of developers per code element.

We utilized the results from the case study to develop new concepts that can reduce the effort involved: in transitioning from one worker to the next (through the Multimind groupware), in managing the tasks among the peer group of workers (through the notion of Composite Persona); in annotating the code (through voice annotations); and in the overall management of the decentralized software development process (via CPro).

In sum, this paper analyzes the characteristics of the 24-hour knowledge factory that utilizes three or more collaborating centers located at carefully selected time-zones to work during daytimes in their respective countries. The efficacy of such a work environment is evaluated by creating a set of 11 hypotheses that are tested in a controlled field experiment involving one collocated team and one distributed team. The results show that the introduction of
spatial and temporal separations between workers implies a corresponding introduction of new challenges; these can be overcome – and even leveraged – for strategic advantage. Our findings suggest that firms can apply the 24-hour knowledge factory model to transition from a service provider framework in which offshoring is a short-term and unilateral cost-saving tactic to a strategic partnership in which offshoring becomes a core component of the global corporate strategy.
REFERENCES


PBS Report: “Platform Sharing Takes The Auto Industry To Another Level”


