Globalizing Software Development

C.J. Meadows
Harvard Business School

In an attempt to provide insight into managing successful international technology collaboration, this article presents a framework to guide managers in dividing and integrating labor and assessing methodological and organizational weaknesses. The analysis draws on technology and management literatures and specifically addresses outsourcing systems development work to emerging economies. Recommendations are provided not only on how to organize international work at present, but also on the characteristics of projects and tasks that are good candidates for international outsourcing, and what capabilities must change in order to move to an international project management methodology involving lower project risk and lower coordination costs.

Like many other industries, software has entered an era of globalization (Kim, et al., 1989; Schware, 1989) due to different factor costs among nations, technological advances (such as improved telecommunications, greater standardization and modularization of software, etc.), and the presence of multinational competitors that are on the leading edge of global work integration. In addition, outsourcing is on the rise (Khosrowpour, 1995; Apte, 1990), and information technology (IT) work is being outsourced internationally in a manner that less resembles international trade and more resembles integrated, international production.

With these changes, IT managers are having to "internationalize" their management practices in order to coordinate and control their international operations. Unfortunately, they face a dual problem of transitioning towards transnational IT management (a) from their own locally-oriented IT management heritage (Cash, et al., 1992), as well as (b) from the heritage of international business, which has traditionally been conducted more as trade than as integrated, international production (Bartlett and Ghoshal, 1992). Tight coordination and control across national borders was not even attempted until fairly recently because of the lack of communications infrastructure (Ohmae, 1989: 139), and new management techniques for dealing with integrated transnational operations are still being developed.

In the face of these changes, managers need guidance in managing their international efforts and deciding which activities should be outsourced across national borders. To this end, the current article presents a framework for thinking about transnational IT management. The article first outlines some trends that are enabling the changes and which are changing the way IT projects are managed. Then the framework is presented, issues of task partitioning and integration are outlined, and recommendations are provided both for current capabilities and infrastructure and for the future. The article does not address how each partner manages its relationship with the other partner, nor does it address how to choose partners or environments to deal with (national or technological environments). Further, the article is a presentation of logical arguments addressing theoretical issues and future practice, rather than an empirical analysis. Very little data (if any) on these ideas exists, so the current work focuses on understanding IT management methodologies, required changes, and potential future practice.

Although the framework could apply to IT management in general, the article focuses on the specific situation of outsourcing software development to emerging economies. Before delving into the framework, two terms should first be defined: outsourcing and emerging economies. Outsourcing, as it is used in this article, refers to sourcing tasks outside of a work group. For example, if an IT department finds it no longer has the resources or expertise to handle all the demands

Manuscript originally submitted June 29, 1995; Revised November 17, 1995; Accepted November 27, 1995 for publication.

for new computer applications, it might contract with a local third party to provide those services (outsourcing). If, however, the department is under cost pressures or needs expertise it can not find locally, it might contract with a third party halfway around the world in an emerging economy (international outsourcing) such as India or the Philippines (see Meadows, 1995, for data on their software industries).

The term "emerging economy" is used here, instead of "less developed country" (LDC). Although "less-developed country" is a popular term, it de-emphasizes the substantial civilized history of the nations referred to and the economic development already achieved. Further, the terms do not focus on the critical issue at hand, i.e. the emergence of these nations as global economic powers. Likewise, "industrialized nation" will be used instead of "more developed country" (MDC) in order to focus on industrialization as an economic phenomenon, and in order to use a comparable term.

New Developments Enabling Change

New systems development methods and tools are enabling the outsourcing and IT globalization trends mentioned above. In the field of software development, near-term trends (Marciniak, 1994) point toward enhanced methods for design, more standardized procedures, better ways of evaluating designs, and rising levels of design quality and overall system quality. Defining requirements will continue to be problematic. In the long-term, the discipline should mature along the same lines as the engineering profession, making systems development more a science than an art. In the meantime, however, it is an art and will continue to be for the foreseeable future.

Trends that appear favorable for outsourcing to emerging economies include greater modularity (such that pieces of systems can be developed off-site more easily), standardization of programming languages and environments (reducing the learning curve and hardware and software investments), open systems (allowing for multiple development environments to be used), rise of client-server and other non-mainframe environments (reducing capital investment requirements in capital-poor nations), rise of more reliable means of defining user requirements (favorable to emerging economy software developers if they can gain access to them and use them effectively), easier transportability of software over international telecommunications facilities (which are improving at a rapid pace), rise of Integrated Project Support Environment (IPSE) systems (which provide a coordinated set of software engineering and management tools), and the rise in outsourcing in general.

Tools/techniques such as prototyping and Rapid Application Development (RAD) that improve user-designer interaction may prove invaluable to overseas outsourcers for improving their interaction with users and producing designs that are more reliable. However, system requirements will probably remain generally unstable, and overseas outsourcers will have to contend with problems of cross-cultural communication, knowledge of local business practices, and the need for rich media (e.g. face-to-face) during relationship-building and unstructured problem-solving tasks.

Not so favorable for systems development outsourcing are trends which increase the productivity of industrialized-nation systems developers and focus systems development activities on the user-developer interface (the weak link in overseas outsourcing, as discussed below), including code generation and reuse, use of higher-level languages (e.g. 4GLs+, OOPS, etc.), enterprise modeling, end-user computing, and increased use of packages (although they frequently must be tailored).

Will new methods and tools make the need for outsourcing to emerging economies obsolete (such that all the research being done in this area, including this article, is needless)? The answer is "No." First, new methods — even if they do reduce overall labor demand by making systems professionals more productive — take on average 18 years to roll out into the industry (Gibbs, 1994). In the meantime, there is a worldwide shortage of 1 million systems-builders, and by the year 2,000, it is estimated that Japan alone may need about one million more software engineers and programmers (Mijares, 1992).

Second, in the face of a growing software labor shortage, most companies have continuing backlogs of new applications and maintenance, and there will always be a need for at least some technical work which can be readily outsourced overseas. Software professionals are increasingly forced to maintain software rather than develop it (60 - 80% of software budgets and 50% or more of corporate I/S staff are allocated to software maintenance) (Kim, et al., 1989; Keen, 1991), furthering the maintenance backlogs and the shortage of labor for systems development. Emerging economies can do a large part of the maintenance and at least some of the development work.

Third, a floating global resource of knowledge workers from both industrialized nations and emerging economies is arising, with the expertise necessary to serve clients in a variety of locations worldwide (Aharoni, 1993). MNCs increasingly need "local" expertise in multiple sites (including emerging economies), and international business holds great potential for at least some degree of homogenization (see, for example, Porter, 1986, and Bartlett, Doz, & Hedlund, 1990). Indeed, globally used information systems may need to be globally developed (on globally used information systems, see, for example, Deans & Kane, 1992, and Roche, 1992).

Framework

Dividing tasks among the groups that participate in an international systems development project is, at its most basic level, a problem of division of labor. Understanding how to organize international outsourcing projects requires addressing three questions:

1. ***Division of Labor Theory***: Without regard to capabilities, how should the work be organized?

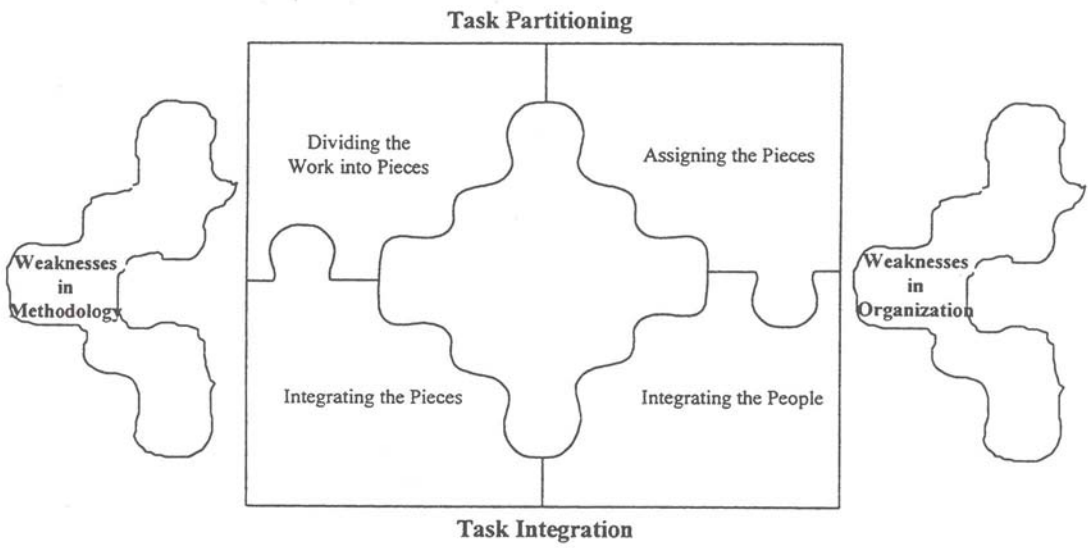


Figure 1: Division of Labor Framework

2. ***Division of Labor Practicalities:*** Capabilities are not unlimited. What can emerging-economy professionals do (now and in the future)?
3. ***Division of Labor Solutions:*** How do #1 and #2 meld together into a practical managerial solution (now and in the future)?

Fundamentally, division of labor involves (a) design of work, or task partitioning, and (b) task integration. As shown in the framework, the first item, task partitioning, involves two activities -- (a) dividing work into pieces such that interdependencies are reduced, and (b) assigning the pieces, based on tightness of communication between the parties performing interrelated work (capabilities are, of course, a prerequisite, and they are addressed below). The second item, task integration, involves (a) integrating the tasks by establishing how the task outcomes will be assembled to form a unified deliverable, and (b) integrating the people, by enhancing the communication capabilities and inter-group cohesion of the groups that perform the work.

As indicated in the framework, these management elements "hold together" the work activities and provide a context for their successful completion. However, these elements cannot ensure success. The underlying principle that managers must not ignore is to *avoid matching methodological weaknesses with organizational weaknesses.* Compounding one with the other can spell disaster. For example, as discussed more fully below, if tasks are highly inter-related (a methodological weak point), and they are assigned to groups that do not interact well either through personal conflict or through lack of communication infrastructure (an organizational weak point), the final deliverable will be in danger. To manage this way is like putting identical pieces into a puzzle and expecting them to mesh with each other and fill the puzzle vacancy (see Figure 1). Instead, managers must attempt to supplement weaknesses in methodology with strengths in organization, and vice versa.

Software Development Approaches	Division of Labor
Waterfall	Time
Phased	Module
Evolutionary	Time or Module
Spiral Time (two-tiered)	
Rapid Prototyping	Module + Other Approach
Disciplined Evolutionary	Mix

Table 1: Software Development Approaches and Division of Labor

Task Partitioning

How does the framework apply to emerging-economy outsourcing? In this section, the top half of the framework will be explored (dividing and assigning tasks) specifically as it relates to international software development projects. In the next section, the bottom half (integrating tasks and people) will be examined.

Dividing the Work into Pieces: Time vs. Module Division of Labor

General approaches to the software development process and the manner in which they divide labor are presented in Table 1 (based on Marciniak, 1994). Rather than describe the approaches in detail here, their underlying division of labor approaches are presented. Critical danger points for international outsourcing, based on these characteristics, are to be examined later.

The well-known *Waterfall* method is basically a division

of labor by time, from requirements specification to top-level design to detailed design to coding, unit test, integration, and user acceptance.

The *Phased* approach divides work by module. A "module" denotes some division related to functions (such as accounting, customer order cycle, inventory, etc.) or some other logical division (such as subsidiary locations, product groups, etc.). As in the other approaches described here as "modular," the highest-level division of labor is modular, but within modules, labor may be divided according to time (e.g. by Waterfall stages). For the current analysis, only the highest-level division of labor is addressed because it would be used to divide work among the groups involved (as discussed below).

The *Evolutionary* approach is a developmental model in which a basic system is put in place (by one of these approaches listed), and work is then divided into growth phases, which could entail minor changes to an array of systems components, added functions, changing capabilities, etc. The evolutionary approach, divided by "growth phases" could either (a) evolve over the systems life cycle (time), with user needs changing in a pattern that do not relate to function, or (b) involve new functions and pieces (business locations, new products or services, new core concept of the business, etc.) that can be called "modules."

The *Spiral* approach is an attempt to combine the best of the Phased and Evolutionary approaches and proceeds through four loops:

1. System requirements and design,
2. Software requirements,
3. Top-level design, and
4. Detailed design, code, test, and delivery.

Within each loop, four steps are taken:

1. Determine objectives, alternatives, and constraints,
2. Evaluate alternatives, perform risk analysis, and create prototypes, simulations, models, and benchmarks,
3. Develop and verify the product of the current loop (one of the four loop titles above), and
4. Plan the next phase.

In short, the Spiral divides work by *time*, like the Waterfall method, but does so in a two-tiered fashion (stages within stages).

Rapid Prototyping, although nominally a separate approach, is mainly a method of specifying user requirements (although some code may be generated, depending on the tools used). In essence, work is divided by module (for instance, by screens grouped into functions or product/service groups, for on-line systems), clustered around user needs or activities. Once user needs are identified in this manner, the ultimate system may be installed via Waterfall, Phases, or with any of the other approaches mentioned here.

Finally, the approach most often taken by sophisticated developers of large systems can be called *Disciplined Evolutionary*. Basically, it makes use of all the other approaches and has the advantage of flexibility (methods can be used or not used when judged appropriate).

At its most basic level, systems development approaches, then, seem to be divided by either time or module. *The weaknesses in the time-based approaches reside primarily in the transition between phases. The weaknesses in the module-based approaches, on the other hand, reside primarily in the links between modules.* These weaknesses will be re-addressed below in relation to the organizational weaknesses that arise when multiple groups are involved in generating software systems and work is divided by time or module among the groups.

Assigning the Pieces

Regarding task assignment, von Hippel (1990) provided a clear and effective example of two-firms — a design firm and a manufacturing firm — and asserted that assigning tasks to them depends on the most critical task interdependency. For example, if the more critical interdependency resides between design of part A and design of part B, then the design of A and design of B should be performed by the same firm (see Figure 2). If, however, the greater interdependency resides between design and manufacture of part B, then the manufacturing firm should both design and manufacture part B (see Figure 3).

Since design and manufacturing are essentially simultaneous for custom software development, the critical interfaces

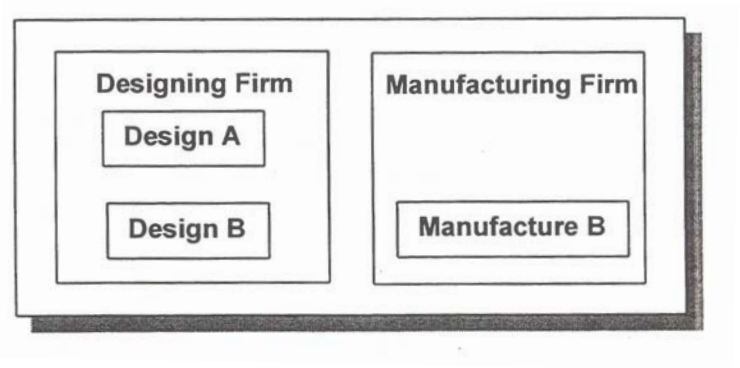


Figure 2: Division of Labor Example: Critical Interdependency Between Design of A and Design of B

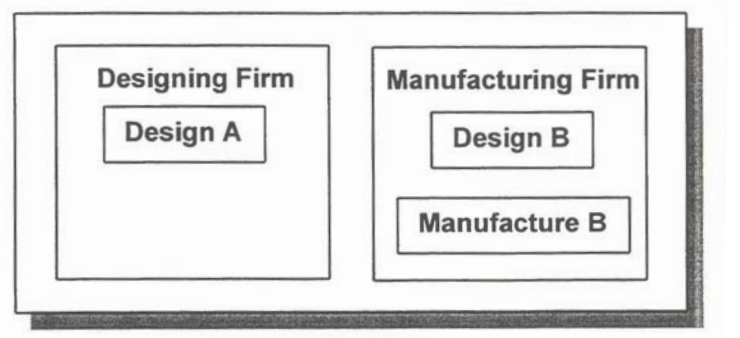


Figure 3: Division of Labor Example: Critical Interdependence Between Design of B and Manufacture of B

in the international outsourcing situation are the interfaces between modules (for module-based division of labor) or stages (for time-based division of labor). Wherever the greater interdependencies reside (between specific modules or specific stages), managers must make sure that organizational integration will support those interdependencies. For example, where modules A and B are critically interdependent, they should be performed in a highly integrated setting (e.g. within a single firm). Similarly, where stages A and B (e.g. design and coding) are highly interdependent, they should be performed in a highly integrated setting (e.g. within a single firm).

In his example, von Hippel assumed that integration is more difficult across firms. Organizational integration may be best understood, however, in a broader context than that of inter-firm coordination. The root of his analysis depends on identifying where communication and integration are weakest. Although this may usually be across firm boundaries, it can also be across discipline/functional boundaries such as users vs. an I/S department (see Schein, 1992, on the different cultures of IT and management, and other literature on the problems of user-designer integration) or across national boundaries within the client or the outsourcing firm (such that a single firm is not well integrated between international offices) or, indeed, across some other boundaries, depending on the situation. Thus, managers must examine integration within firms as well as between them and assess the boundaries along multiple dimensions (organizational, discipline, national, technological, cultural, etc.).

Assigning work is not only a matter of methodological and organizational integration, however. Assignments must, of course, depend on capabilities. Capabilities of emerging economies include not only the managerial and technological capabilities of the firms, but also national characteristics such as regulation, telecommunications infrastructure, and other infrastructure enablers. Taking into account both firm-level and national-level characteristics, Kim et al. (1989) identified competitive activities for emerging economies to include programming, systems analysis (low supply of analysts, however, in some nations, as well as problems in end-user communication), technical writing (again, low supply in some nations), and hardware engineering. They also found that emerging-economy systems developers had better attention to detail and more group orientation.

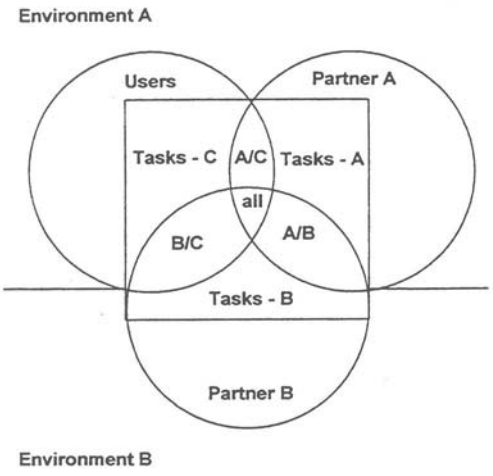
Further, a number of firms are finding that "follow-the-sun" systems development can provide a powerful advantage to overseas outsourcers. Because of differences in time zones, a systems professional in India can batch and send questions at the end of the day to a client in the U.S. who, because of the time difference, is just getting into work. Questions are answered overnight (Indian time), and overall project schedules can be reduced because the work is carried on around-the-clock. Likewise, a U.S. client can request system changes at the end of the day (U.S. time) and receive a new version of software in the morning (or in one business day instead of two). Although an advantage for questions and small fixes, the time zone difference is a hindrance when questions need to be answered immediately (only next-day answers are possible), workers at both sites need to collaborate on unstructured problems, off-site systems developers need to learn users' jobs, and when regular team meetings need to be held by video-conference across a 12-hour time difference (Meadows, 1995). "Follow-the-sun" is a mixed blessing.

On the down-side, emerging economies face limited availability of capital (for purchasing hardware and software tools), limited availability of technical expertise (in some nations), cultural and language barriers (communication protocol and meanings of words vary from nation to nation), and differences in managerial practices and expectations (such as status reporting). Non-competitive activities identified by Kim et al. (1989) included communication with end users, effective intra-organizational communication, and project management (cheaper, but extremely limited supply and weaknesses).

Limited availability of hardware and systems software and tools were also cited as major problems of emerging economies which hampers the activities they can perform. Management capability was found to be hampered by weaknesses in establishing performance-based rewards and calling for well-specified milestones and deliverables. Indeed, Yourdon (1993) mentioned a deficiency of large-project management capability in India (which is changing), and Palvia and Hunter (1994) confirmed that there can be a significant time lag in transferring new management techniques, methodologies, and tools from industrialized nations to emerging economies.

Kim, et al. (1989) deemed the scarcity of performance-based rewards and well-specified milestones a "cultural weakness" in management, but it may simply be the result of overseas outsourcers' position low on the project management learning curve. In essence, some firms may not have yet converted the task of systems-building from a "novel innovation" task into a "routine innovation" task with established rewards, management methodologies, and deliverables. (See von Hippel, 1990, on novel and routine innovation tasks and their management methods.)

It is important to question the reason for this state of affairs because culture will not change quickly, whereas the transition from "novel" to "routine" innovation management can. The areas Kim, et al. (1989) deemed not shiftable to emerging economies included: conceptualization, documentation, installation, and support. They cited cultural differences, limited local markets, insufficient R&D, inadequate telecommunications facilities, and delayed access to new technologies as barriers. Of these, cultural differences will probably be the slowest to change and will continue to impede communication with Users. Work is going on in all other areas, however, and they have improved significantly over the past five years. Although not yet the industry norm, start-to-finish projects have been conducted between industrialized-nation clients and emerging-economy outsourcers with stunning success, sometimes in high-technology and highly user-interactive areas (Meadows, 1995).



Note: Environments A and B in international outsourcing are different nations.
Circles represent groups. The inner box represents the systems development project, divided into tasks.
Partner "A" is an outsourcer in the Users' nation or internal I/S department in user A's nation and firm.
Partner "B" is the overseas (international) outsourcer.
Users perform Tasks "C."

Figure 4: Overseas Outsourcing Division of Labor

Integrating the Pieces and the People

Von Hippel (1990) did not define integration between groups more than to say that it occurs where communication is frequent. Here, another idea must be added: communication richness. Although communication frequency is important for the development of cooperative relationships (Axelrod, 1984), information availability, and idea generation (Sproull & Kiesler, 1991), rich media (such as face-to-face) are necessary for decision-making (Nohria and Eccles, 1992) and relationship-building (Sproull and Kiesler, 1991, dispute this claim). In fact, there has been a stream of literature on optimal choice of systems design tools and media for various problem-solving tasks during I/S development. (See Palvia & Hunter, 1994 for a brief review of this literature.)

In the overseas outsourcing situation, integration would typically be a matter between three groups (see Figure 3):

1. Users,
2. Partner A, which might be an internal I/S group and/or a domestic outsourcing firm (for partially or fully outsourced I/S), and
3. Partner B: the Overseas Outsourcer.

Though potentially relevant, the multiple outsourcing case, in which I/S is outsourced to multiple internal I/S groups and/or multiple outsourcing firms, will not be addressed here, beyond noting that the number of involved parties and coordination problems can rise alarmingly.

The weakest link is between Users and Partner B. Their relationship must cross three boundaries: function/profession, firm, and nation (yielding not only cross-cultural problems but more mundane problems of inadequate telecommunications facilities and differences in time zones). The User - A and A - B links are impossible to assess without knowing the particular situation. Users and A, although they do not share the same function/profession, do share the same nation and may share the same firm. A and B, on the other hand, share the same function/profession and may share the same firm, but do not share the same nation. Whereas Users and A (or A and B) can rely on communication links via profession (similar jargon, common paradigms and tools, common professional culture) or via nation (language, culture, face-to-face media) or via firm (organizational jargon or culture, common processes and systems), Users and B can rely on none of these communications links.

What is proposed here is that the international systems development manager must first identify the critical task interdependencies in the methodology(ies) in question. Next, identify the critical weaknesses in integration among partners (see organizational integration literature, e.g. Simon, 1976, Lawrence and Lorsch, 1986, and others). Then, match methodology and partners/organization to achieve maximum "fit," such that critical task interdependencies are not confounded by organizational weak points. Then, where weak points exist (especially in both methodology and organization), improve organizational communication and integration. To ignore these issues may result in:

1. Increased project risk (perhaps dangerously), and
2. Increased coordination costs (including management time, travel, telecommunications, dual hardware and software environments, etc.; see Brooks, 1975, on how coordination costs can destroy the potential productivity of additional manpower).

In the time-based approaches (Waterfall, etc.), the critical weakness is at the juncture between stages (requirements analysis, design, programming, etc.). Unfortunately, when dividing up international outsourcing work via waterfall stages, the stage transition (weak-point) is also the organizational transition (weak point). This results in matching methodology weaknesses with organizational weaknesses.

How critical is this weakness? According to Thompson (1967) (cited in Larsson and Bowen, 1989, and Mintzberg, 1979), interdependence can be classified as pooled, sequential, or reciprocal. In pooled interdependence, each part renders a contribution to the whole, and the proper coordination mechanism is standardization. With sequential interdependence, the output of one party becomes the input of another, and the proper coordination mechanism is planning. With reciprocal interdependence, the output of each becomes the input for the others, and the appropriate coordination mechanism is mutual adjustment.

Although the time-based methodologies are theoretically sequential (with the output of one stage, such as design, being the input of another, such as coding), systems develop-

ment in reality is reciprocal (inconsistencies are identified, for example, during coding and require redesign) (Walz, et al., 1993):

> The traditional approach to software development recommends that these topics be addressed in sequence....[however,] it is clear that these steps were not addressed in sequence, they were not independent of one another, and they did not appear to have clear starting and ending points. (p. 67)

Why is this significant? Because the very fact that the process is reciprocal rather than sequential indicates that a higher level of coordination is necessary (mutual adjustment over planning and following the sequential methodology) and indicates that rich communication and integration are crucially needed among the parties involved.

In order to avoid depending on the weak integration between the Users and Overseas Outsourcer, the situation can be managed with a "front-office" of workers at the Users' location and a "back-office" of workers overseas. Work could be divided between "front-office" and "back-office" via the Waterfall methodology such that more user-interactive tasks are handled by the "front-office" (see Larsson & Bowen, 1989). The disadvantage in doing so, however, is that overall task interdependency is quite high, and the critical interfaces between stages still have to be managed over organizational and national boundaries (between Partner A and Partner B).

In the module-based approaches (Phased, Disciplined Evolutionary, etc.), overall task interdependency is reduced over the time-based approach (a major advantage). The interlinkages between modules will have to be managed carefully over organizational boundaries, but this is certainly workable. However, the primary disadvantage for the overseas outsourcer is that the early phases of any module require rich communication between users and software developers (mutual adjustment, as described above). This will be difficult to do when crossing boundaries of function/profession, organization, and nation (as in the User - B relationship) and will endanger the success of the modules they are assigned.

Thus, in one approach (time-based), overall task interdependency is high, but the strongest integration links are used. In the other approach (module-based), overall task interdependency is lower, but mutual adjustment for part of the system is necessary across the weakest organizational link (User - B). Two assessments must therefore be made:

1. The capabilities of the various partners must be assessed, and, if capabilities permit,
2. The risks in the time-based approach (from increased interdependency) must be weighed against the risks in the modular approach (from relying on the weak User - B relationship rather than User - A and A - B). The modular approach risks will vary greatly according to the specific relationships among Users, Partner A and Partner B, so they must be assessed for each individual situation and then compared with the time-based interdependency risks (which are more constant across situations).

Waterfall Stage	Performed By ...
System Feasibility	Users and Partner A
Validation	Users and Partner A
Software Plans and Requirements	Users and Partner A
Validation	Users and Partner A
Preliminary Product	Design All; Mainly Users and Partner A
Verification	All; Mainly Users and Partner A
Detail Design	All; Mainly Users and Partner A
Verification	All; Mainly Users and Partner A
Code	A and B; Mainly Partner B
Unit Test	A and B; Mainly Partner B
Integration	A and B; Mainly Partner B
Product Verification	All; Mainly Users and Partner A
Implementation	All; Mainly Partner A
System Test	All; Mainly Partner A
Operations and Maintenance	All; Mainly Users and Partner A
Revalidation	Users and Partner A

Table 2: Division of Labor Example

Recommendations

Current Division of Labor

Overall, the detailed, structured systems development tasks in the middle stages of the systems life cycle (programming and testing, etc.) have been the most popular candidates for emerging economies to work on and are especially good candidates for the early stages of outsourcing relationships (when client confidence is built along with the software), resulting in a time-based division of labor. Tools, methodologies, and management techniques (e.g. an Integrated Project Support Environment) need to be agreed upon and transferred before commencing projects in order to reduce both coordination costs and project risk (by improving integration).

Table 2 shows a fairly typical division of labor employed by several firms in the exploratory study for this research, and used successfully by others reported in the literature.

Because designs evolve during the coding and testing process, and because Partner B's knowledge of the coding, testing, and integration can be very important for later stages, selected members from the overseas outsourcing organization should be involved from design to maintenance (see Table 2), not because they will hold primary responsibility for these tasks, but rather, for reasons of group learning, information flow, and integration (found in the empirical research of Meadows, 1995; in the word of Walz, et al., 1993):

> Knowledge acquisition, sharing, and integration are all activities that enable the software design team to learn what it needs for producing an appropriate design. Seldom are these activities explicitly accounted for in the design phase.... [We cannot overemphasize] the importance of including relevant team members from the beginning of the project. If new members (and their relevant expertise) are added after the group has come to closure in its learning phase, the group may be reluctant to deal with the new knowledge they bring to the team. Thus knowledge at this point may not be incorporated easily into the group's work (pp. 69 - 70).

What tasks are good candidates for overseas outsourcing? In general, tasks that require less integration with the user are good candidates, as well as structured tasks involving explicit knowledge, in order to reduce coordination costs and risk. Amount of structure in a task will vary with the particular problem at hand and a particular technology (with code generators, 4GLs, and OOPS, for example, a bigger proportion of time is spent on problem definition and modeling than with earlier technologies). Amount of structure will also change over the life of a task, from (a) problem recognition (very unstructured) to (b) analysis & design to (c) programming, testing, and implementation to (d) maintenance and conversion (very structured) (Apte, 1990).

To reduce overall project risk, relatively small, noncritical tasks involving technology and functionality familiar to the overseas developer are good choices to begin with. However, as experience with a particular outsourcer progresses or where a good track record exists that will minimize fears of mismanagement or insufficient expertise, these risk-reducing recommendations can be relaxed. In fact, it is desirable to do so since increasing the size of the outsourced portion will reduce overall project costs further, as long as tasks do not raise coordination costs unreasonably. Increasing the portion of outsourced work will involve more coordination, but proportionally not as much as the initial work, since some of the coordination costs are related to the tasks (these coordination costs rise as more tasks are sent over), and some are related to building the initial relationship (these coordination costs do not rise with more tasks). At each stage, additional coordination costs must be weighed against potential cost savings.

What projects are good candidates for overseas outsourcing? In general, a cost - benefit analysis should be performed when evaluating the overseas outsourcing option, and the planned savings must be compared with increased risk. Costs that go up include management time for coordination and relationship-building, travel costs, hardware and software costs (for dual environments), telecommunications costs, etc. Costs that go down may include labor and rent. Risks increase (a) because of the additional complexity of managing an international project and (b) because one of the parties— the overseas outsourcer —is not a fully understood entity.

In general, although larger projects have more risk (more things can go wrong, and with bigger consequences), the project must be large enough to repay the investment in relationship-building costs. Stable requirements or a fixed requirements "cut off" are crucial. Mature, reliable technology is a good candidate to go overseas because of the reasonable risk level, availability of older technologies, and abundance of labor willing to work on them. Since both parties will, in the initial stages at least, have enough problems managing the dual environments and outsourcing relationship, both parties should have some experience with the technology. However, as circumstances change, these two technology-related recommendations will also likely change.

Because of the challenges integrating a project team across national and organizational boundaries, structured projects involving explicit knowledge are good candidates, since they do not require as much integration and rich communication as unstructured tasks involving implicit knowledge. Examples of such projects are those late in the systems lifecycle (e.g. maintenance of existing systems) and those which require less interaction with users (e.g. work which is more technical in nature, such as platform migration). However, as before, once the partners have established a relationship and methods of working together, this recommendation can (and probably should) be relaxed, and examples can be found of highly successful projects which were communication-intense, unstructured, and interactive (Meadows, 1995).

Evolution Over Time

These tradeoffs and recommendations will change as the partner capabilities and inter-group integration change. Telecommunications infrastructures, and technological, functional, and project management capabilities of overseas outsourcers are growing quickly. Other infrastructures, such as intellectual property rights or other legislation, will probably change a bit more slowly, but action is being taken there, too (Meadows, 1995). Cultural characteristics will change slowly, but cross-cultural adaptive skills can be built relatively quickly.

What will probably never change is the integration problem between Users and Partner B, although it may subside somewhat due to increased use of rich media (e.g. videoconferencing) and systems design tools (e.g. prototyping), and also because of the rise of global cultures and business systems. Indeed, Kim et al. (1989) predicted that industrialized-nation firms would retain an overwhelming advantage in the software design phase because of access to users, widely diffused analysis skills, and cultural facility in one-on-one user interaction.

Nonetheless, the analysis provided here suggests that systems builders should shift to a module-based division of labor from the time-based division currently encouraged by emerging-economy capabilities and the weakness of integration between users and overseas systems-builders. Such an action should not be taken, however, until capabilities permit, and User - B integration improves such that the rise in interdependency risk in the time-based approach (over the modular approach) outweighs the risks involved in relying on the User - B relationship rather than User - A and A - B. In order to make the shift, which should reduce both overall

project risk and costs (due to an increased portion held overseas), industrialized-nation managers must focus on improving User - B integration with rich communication media and tools for software development and project management. Emerging-economy managers must focus on (a) expanding capabilities to cover project management and the early and late phases of systems development, and (b) improve communication with Users by acquiring design/communications tools (such as prototyping) and improving communications infrastructures (such as common, dedicated communications links which can be used for electronic mail with clients). There is a limit to how much communications can be improved, however, and systems builders can develop expertise in only a limited number of business environments and technologies. The ultimate solution may reside in improvements such as these coupled with local partnering where needed.

Summary and Conclusions

Faced with globalization and outsourcing trends, managers are "transnationalizing" their IT management practices. To help guide managers in this situation, a framework is presented here which divides the relevant issues (at a highly abstract level) into task partitioning (dividing and assigning tasks) and task integration (integrating tasks and people). One of the key insights in the analysis is that when carrying out these partitioning and integrating activities, managers must identify methodological weak points (where work is likely to go wrong) and organizational weak points (where there is poor integration or lack of capability) and avoid matching them. To confound methodological weakness with organizational weakness could spell disaster.

When organizing systems development work across borders, tasks can be divided by time or module. It is proposed here that time-based division of labor involves greater integration risk overall, but does not rely on the critical organizational weak point -- integration between the User and the Overseas Outsourcer. Module-based division of labor involves lower integration risk in general, but does rely on the organizational weak point. It is recommended here that there should be a shift from time-based to module-based division of labor as the emerging-economy outsourcing industry matures in general and as specific client-outsourcer relationships mature.

It is hoped that this extension of von Hippel's inter-firm task partitioning ideas to an inter-functional, inter-firm, international situation has been useful and that the addition of communication richness to his definition of organizational integration can encourage useful insight. As the practice of international systems development and the field of systems engineering progress, it will be necessary to ask whether the principles of industrial engineering and international manufacturing can be applied to software development, which is (a) a service, (b) a customized product, and (c) an intellectual product. Perhaps these ideas are at least a good start.

References

Aharoni, Yair, ed. (1993) *Coalitions and Competition: The Globalization of Professional Business Services.* London: Routledge.

Apte, Uday. (1990). Global Outsourcing of Information Systems and Processing Services. *The Information Society*, 7, 287-303.

Axelrod, Robert. (1984). *The Evolution of Cooperation.* Basic Books (Harper Collins).

Bartlett, C. A., Doz, Y., and Hedlund, G., eds. (1990). *Managing the Global Firm.* London: Routledge.

Bartlett, Christopher A., and Sumantra Ghoshal. *Transnational Management: Text, Cases, and Readings in Cross-Border Management.* Homewood, IL: Irwin, 1992.

Brooks, Frederick P., Jr. (1975). *The Mythical Man-Month: Essays on Software Engineering.* Reading, MA: Addison-Wesley.

Cash, James, F. Warren McFarlan, James McKenney, and Lynda Applegate. (1992). *Corporate Information Systems Management: Text and Cases.* 3 ed. Homewood, IL: Irwin.

Cyranek, Gunther, and Bhatnagar, S. C. (1991). *Technology Transfer for Development: The Prospects and Limits of Information Technology.* New Delhi: Tata McGraw-Hill.

Deans, P. Candace, and Kane, Michael J. (1992). *International Dimensions of Information Systems and Technology.* Boston: PWS-Kent Publishing Co.

DeMeyer, Arnoud. (1991). Tech Talk: How Managers Are Stimulating Global R&D Communication. *Sloan Management Review*, Spring, 49-58.

Enright, Michael J. (1993). *The Determinants of Geographic Concentration in Industry.* Harvard Business School working paper 93-052.

Gibbs, W. Wayt. (1994). Software's Chronic Crisis. *Scientific American*, September, 86-95.

Humphrey, Watts S. (1989). *Managing the Software Process.* New York: Addison-Wesley.

Jones, Capers. (1994). *Assessment and Control of Software Risks.* Englewood Cliffs, NJ: Yourdon Press.

Keen, Peter G. W. (1991). *Shaping the Future: Business Design Through Information Technology.* Boston: Harvard Business School Press.

Khosrowpour, Mehdi. (1995). *Managing Information Technology Investments with Outsourcing.* Harrisburg, PA: Idea Group Publishing.

Kim, Chai, Westin, Stu, and Dholakia, Nikhilesh. (1989). Globalization of the Software Industry: Trends and Strategies. *Information and Management*, 17, 197-206.

Krepchin, Ira. (1993). When Offshore Programming Works. *Datamation*, July 15.

Larsson, Rikard, and Bowen, David E. (1989). Organization and Customer: Managing Design and Coordination of Services. *Academy of Management Review*, 14, 2, 213-233.

Lawrence, Paul R., and Lorsch, Jay W. (1986). *Organization and Environment.* Boston: Harvard Business School Press.

Marciniak, John J., ed. (1994). *Encyclopedia of Software Engineering.* New York: John Wiley & Sons, Inc.

Meadows, C.J. (1995). Software Development Capabilities -- A Comparative Analysis: India vs. the Philippines. *Information Technology and Socio-Economic Development.* Mayuri Odedra-Straub, ed. Nashua, NH: Ivy League Publishing Ltd., 203-216.

Meadows, C.J. (1995). *Globework: Creating Technology With International Teams.* Doctoral thesis (in progress). Boston: Harvard Business School.

Mijares, Roy. (1992). The Philippines as a Software Production Base. *RIM: Pacific Business and Industries*, 2, 33-49.

Mintzberg, H. (1979). *The Structuring of Organizations.* New York: Prentice-Hall.

Nohria, Nitin, and Eccles, Robert G., ed. (1992). *Networks and*

Organizations: Structure, Form, and Action. Boston: Harvard Business School Press.

Ohmae, Kenichi. (1989). Planting for a Global Harvest. *Harvard Business Review*, July-August, 136-144.

Palvia, Shailendra, and Hunter, M. Gordon. (1994). An Evaluation of Information Systems Development Practices in Singapore. *Managing Social and Economic Change with Information Technology: Proceedings of the Information Resources Management Association International Conference.* Harrisburg, PA: Idea Group Publishing.

Palvia, Shailendra, Palvia, Prashant, and Zigli, Ronald M. (1992). *The Global Issues of Information Technology Management.* Harrisburg, PA: Idea Group Publishing.

Porter, Michael E., ed. (1986). *Competition in Global Industries.* Boston: Harvard Business School Press.

Roche, Edward M. (1992). *Managing Information Technology in Multinational Corporations.* New York: Macmillan.

Rockart, John F., and Short, James E. (1989). IT In the 1990s: Managing Organizational Interdependence. *Sloan Management Review*, Winter, 7-17.

Schein, Edward. (1992). *The Culture of Information Technology and the Culture of Management.* Working Paper.

Schware, Robert. *The World Software Industry and Software Engineering: Opportunities and Constraints for Newly Industrialized Economies.* World Bank Technical Paper No. 104. New York: World Bank, 1989.

Simon, Herbert A. (1976). *Administrative Behavior.* New York: The Free Press.

Sommerville, Ian. (1992). *Software Engineering*, 4 ed. New York: Addison-Wesley.

Sproull, Lee, and Kiesler, Sara. (1991). Computers, Networks, and Work. *Scientific American*, September, 116-123.

Stone, Eric. (1991). Drudge Work Goes Offshore. *Asian Business*, July, 41-42.

von Hippel, Eric. (1988). *The Sources of Innovation.* New York: Oxford University Press.

von Hippel, Eric. (1990). Task Partitioning: An Innovation Process Variable. *Research Policy*, 19, 407-418.

Waltz, Diane B., Elam, Joyce, and Curtis, Bill. (1993). Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration. *Communications of the ACM*, October, 63-76.

Yourdon, Edward. (1993). *Decline & Fall of the American Programmer.* Englewood Cliffs, NJ: Yourdon Press (Prentice-Hall, Simon & Schuster).

CJ Meadows is finishing a Doctorate in Business Administration at the Harvard Business School, with a concentration in Information Systems and International Business. Her dissertation work focuses on international project management, the globalization of services such as software development, and communication management. She specializes in the study of Asian emerging economies and is a visiting professor at the Asian Institute of Management in Manila, Philippines. She has taught in the U.S. and abroad, worked as a computer systems consultant, and has a CPA and CMA.

0 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/article/globalizing-software-development/51273

Related Content

Dispersion or Agglomeration: How Digital Service Trade Barriers Induce China's Outward Foreign Direct Investment?
Ziwei Liuand Yibing Ding (2023). *Journal of Global Information Management (pp. 1-17).*
www.irma-international.org/article/dispersion-or-agglomeration/321177

European Electronic Service Infrastructure Building - Drifting into the Future?
Ake Gronlund (2002). *Global Perspective of Information Technology Management (pp. 77-87).*
www.irma-international.org/chapter/european-electronic-service-infrastructure-building/19277

Outsourcing Information Technology: The Role of Social Capital
James J. Hoffman, Eric Waldenand Mark L. Hoelscher (2008). *Global Information Technologies: Concepts, Methodologies, Tools, and Applications (pp. 530-537).*
www.irma-international.org/chapter/outsourcing-information-technology/18988

Technology Adoption in Post-Conflict Regions: EDI Adoption in Kosovo after the War
Larry Stapleton (2013). *Global Diffusion and Adoption of Technologies for Knowledge and Information Sharing (pp. 241-261).*
www.irma-international.org/chapter/technology-adoption-post-conflict-regions/72190

Internet Presence and Multilingual Dissemination in Corporate Websites: A Portrait of Spanish Healthcare SMEs
Irene Rivera-Triguerosand María-Dolores Olvera-Lobo (2021). *Journal of Global Information Management (pp. 1-17).*
www.irma-international.org/article/internet-presence-and-multilingual-dissemination-in-corporate-websites/274047