

1 Introducing the phenomenon of global software work

1.1 Introduction

It is 9 a.m. Monday morning and Peter Kelly, Managing Director of Academy Information Systems in Trowbridge, UK, has just arrived at his desk. He sits down to examine the progress on the latest release of Academy software for housing benefits. For three years this development has been outsourced to Mastek, an Indian software company. Kelly consults the Mastek website relating to the project and the 'dashboard' shows relevant indicators of quality, utilization, efficiency and schedule. Subsequently, he meets Sanjay, Mastek's project manager, who updates him on the progress verbally. Part of Academy's project team has arrived for work six hours before Kelly and has already made progress on several programming specifications given to them the previous evening. This is because the majority of the project team live and work in a different time zone, country and culture at Mastek's India development centre in Mumbai. Around lunchtime in the UK, before the Mumbai part of the project team leaves for home, they transfer the completed code to the server in Academy's Trowbridge office. The UK-based Mastek and Academy staff then have time for testing the completed code before incorporation into the beta release of the application. They can then prepare detailed specifications for the India-based team that they will pick up electronically in the Mumbai morning.

This brief story is an insight into the day-to-day life of Global Software Work (GSW), which is the topic of this book. We define GSW as 'software work undertaken at geographically separated locations across national boundaries in a coordinated fashion involving real time or asynchronous interaction'. GSW can thus include work done across global borders through outsourcing, alliances, or subsidiary arrangements. GSW is still an unexplored form of work and is enabled through organizational forms quite distinctive from traditional global arrangements as typified by large multinational corporations. Unlike manufacturing activities and professional services such as consultancies that have been studied in the past, software development in global settings remains empirically largely unexamined. Software development is a knowledge-intensive activity, and typifies work in the 'knowledge' or 'network' society. An analysis of such work in practice can provide interesting insights into the kind and extent of operations that can be effectively conducted in conditions of globalization. GSW takes place within an extremely dynamic and diverse global marketplace that is populated by organizations big and small from countries both developed and developing. The GSW arena is thus unique in that firms need not be fatally handicapped by existing size, and can

potentially make an impact based on their knowledge competencies, ability to leverage technology and the cost advantages they offer. Diversity, complexity and uniqueness are thus inherent in GSW, making it an exciting and relatively unexplored domain of study. Analysis of GSW has implications for different disciplines concerned with such arrangements, including information systems, international management, computer supported collaborative work and organization theory. GSW arrangements are also of concern to policy makers responsible for economic growth and infrastructure development, particularly in developing countries such as India that has benefited greatly from an expanding export oriented software industry.

The aim of this book is to develop an empirically informed understanding of the process of Global Software Alliances (GSAs), the organizational arrangements that are established for the conduct of GSW. The evolution of GSAs are conceptualized within the context of globalization. We do this through the analysis of case studies that allow for an interrogation of various issues in the relationship from a variety of perspectives. Through inter-case comparisons, we seek to develop theoretical and managerial implications that can inform a better understanding about the conduct of the GSW phenomenon. The book can be read on two levels. First, and primarily, it can be treated as a study of globalization, examining specific cases that are both a model *of* and a model *for* globalization. Secondly, our analysis will be of interest to managers and practitioners charged with the task of undertaking GSW who are prepared to go beyond simplistic ‘how to’-type guides and methodologies. The strength of the approach is in the use of case studies to provide an in-depth analysis of particular issues, together with rigorous employment of theory that can help to develop both practical and theoretical implications.

We have structured the book in eleven chapters that may be read in a linear fashion from ‘cover to cover’ or as a reference and resource in the GSW area. Chapter 1 introduces the phenomenon of GSW. Chapter 2 provides an exposition of the theoretical underpinnings of the research approach and chapters 3–9 describe and analyse key themes within the detailed case studies of companies involved in GSW in India. These cases are the result of research undertaken during the period 1995–2000, involving some 200 interviews. Chapters 10 and 11 are concerned with the implications of our analysis at a theoretical and practical level, respectively.

The aim of this chapter is to present the GSW phenomenon in depth and set up the theoretical basis for the subsequent analysis. This phenomenon is shaped by three defining themes relating to the nature of the organizational form that enables such work, the kind of work that is conducted and the complex global trends within which such work is carried out. A detailed discussion of these three themes now follows.

1.2 Organizational forms and GSW

Globalization is a key characteristic of change in many domains at the turn of the twenty-first century. The most visible aspects of globalization have been a bewildering collage

of transformations – increasing religious fundamentalism coexisting with greater secular human concern, development of centres of advanced technology amid regions of poverty and interconnectedness of systems and regions in ways that did not exist before. International business environments and organizational forms are being significantly reshaped as part of a new scenario that has variously been labelled as the ‘new economy’, the ‘digital economy’, the ‘network society’, or the ‘information age’. In these new environments, changes are especially visible in the kind of *organizational forms* being adopted to enable global work. A distinctive and defining aspect of these new forms is the manner in which *space and time* have become the primary medium through which to rethink the nature of the organization (Friedland and Boden 1997). An example of one such new organizational form is the ‘Global Software Alliance’ (GSA), a term we use to describe the nature of organizations established to enable GSW.

A GSA can be conceptualized as a relatively long-term inter-organizational relationship established between the outsourcing organization (the outcomes) and the outsourced organization (the contractor) based in different countries to enable software development in both real time and asynchronous time. This development occurs primarily in shared electronic domains with developers being located in the physical premises of their respective organizations (referred to as ‘offshore’). The opening vignette demonstrates a case of offshore programming and emphasizes the additional dimensions of managing projects across distance, time, language and culture. Offshore arrangements contrast with earlier ‘body-shopping’ (see below) where programmers from the outsourced firm carried out development while being physically located in the outsourcing organization (referred to as ‘on-site’). Taking advantage of the increasing sophistication and capacity of telecommunication links and relatively lower labour costs in the outsourced organization, work in GSAs is done primarily in electronic spaces created through the use of information and communication technologies (ICTs) such as videoconference and email. While the physical travel of personnel between the vendor and contracting organizations can never be completely eliminated, the ongoing quest of both sides is to optimize costs by minimizing travel and finding the appropriate blend between on-site and offshore development. As GSAs seek to find synergies between remote and face-to-face work, time, space, organizational and national boundaries are recombined in novel ways where the experience of ‘here’ and ‘now’ loses its immediate spatio-temporal referents and becomes tied to and contingent on actors and actions at a distance.

Historically, the fortunes of firms in developing countries were seen as tied to the fortunes of those in the developed world. Our research into GSW provides some examples of firms in the developed world whose own fortunes are tied with equal potency to those in the developing world. Prior to GSW arrangements being possible, global work was primarily conducted by large organizations by virtue of their substantial direct investment transcending national borders. Based on their theory of a strategic mentality, Bartlett and Ghoshal (2000) categorized such firms as being *international*, *multinational*, *global* or *transnational*. At one end of the spectrum are international organizations that use their overseas operations in a marginal way, for example, simply to

supply raw materials and marketing contacts to the parent company. At the other end, transnational organizations seek to integrate overseas operations more fundamentally by developing global efficiencies while also creating locally responsive approaches. In between, there is the multinational corporation (MNC) which takes a flexible approach by modifying its practices and products across countries. Managers of transnational organizations adopt a global outlook and seek to develop standardized approaches based on the assumption that there are more similarities than differences across countries. In centralized global companies, foreign units are dependent on headquarters for funds and expertise, but the transnational selectively centralizes some resources at home and some abroad in keeping with the need to respond flexibly to different issues. The transnational corporation is characterized not by structure alone but by formal organization, information systems (IS), culture and values.

Although the Bartlett and Ghoshal typology may still hold in the categorization of different kinds of software firms doing work globally, what is interesting is that these firms are quite different from those that have traditionally operated internationally. Size and ability of the firm to make large-scale investments on infrastructure are no longer terminally limiting factors in whether or not they can undertake GSW. Rapid upgrades in information and communication technologies (ICTs) have reduced the cost of communication and increased the scope of operations so that relatively small companies can potentially have business relationships and can address markets in different geographical domains. Some firms, particularly in such sectors as software, web development and other new media supported by networked and shared IT infrastructures, are capable of competing with larger companies in the global marketplace. Being an Indian or Russian firm is less of a perceived disadvantage and such firms are in fact sometimes actively sought by larger ones by virtue of the knowledge capital they hold, the cost advantages they offer and the potential they provide to serve as a basis to access new markets. Along with large IT companies such as IBM and Microsoft there are many examples of firms who despite being small, are 'born global' and are capable of operating in a multitude of domains and countries (Saxenian 2001).

Saxenian argues that today new transportation and communication technologies permit even the smallest firms to build partnerships with foreign producers and tap overseas expertise, cost savings and markets. Start-ups in Silicon Valley today are often global actors 'from the day they begin their operations' (2001: 5). This multiplicity of networks in which these firms operate makes it difficult to categorize them on single dimensions of domains of work or countries of operations. They are better understood on their ability to develop and sustain *networks* that enable the flows of information, expertise, knowledge, and capital. Networks allow these firms to switch rapidly between local and global domains and build competence in different functional areas and markets. For example, Arrk, a small UK-based software house located in the University of Manchester Science Park employing only forty people, has the majority of its programmers in India and an international portfolio of customers. Cisco Systems has defined its core competence as product innovation, marketing/customer service

and business relationship management. It delegates the rest, such as manufacturing assembly and product configuration, to its partners.

In operating these multiple networks, software firms deal not only with the strategic issues of whether or not and where they should globalize, but also with day-to-day operational issues including the creation of infrastructure, defining management processes and developing language and cultural understanding. Global projects have independent, autonomous links, and modules of work are distributed and coordinated using ICTs across wide physical and cultural distances. ICTs help both to intensify and redefine the nature of interactions across these different nodes which are not only confined to large organizations but also take place at the level of small firms and work teams. For reasons of geography and history, such as physical separation of different units and limited prior relationships of partners, these networks can comprise multiple short-lived global software teams (Carmel 1999). This is fundamentally different from firms composed of relatively autonomous units located in several countries as described by Bartlett and Ghoshal. However, the GSA relationship between the outsourcing and outsourced firms can take on different forms including joint ventures (JVs), vendor contract relationships and fully owned subsidiaries. New relationship models are also emerging: broker companies, for example, build databases of users and providers of outsourcing services and match firms based on predefined criteria. Some of these broker firms try to give more value than mere matching and provide project management services once the relationship is established. Another example is the 'hub' model where, for example, a Japanese firm may use its Singapore subsidiary through which to outsource to India. This model is used in an attempt to cost effectively and bridge some of the language, cultural and infrastructural gaps that would exist if work were carried out in India.

The organizational model adopted directly influences the pricing basis, that can vary from 'time and materials' to 'turnkey' or 'fixed price'. While in a time and material model development is priced on the programmers' time spent, in the other two cases, the basis is the estimated value of the whole project. The basis adopted has significant implications for intellectual property (IP) issues and the project control measures that need to be adopted. Where commitment in the relationship is not long term, and the aim is not to contract out new and core technologies, vendor contracts rather than JVs and subsidiary arrangements might be preferred. Relationships operate over different levels of a *trust continuum* (Heeks 1995) that is shaped by various considerations, including the length of the relationship, the kind of projects being done, the material investments made by both parties and the management capabilities to deal with the complexities of time, space and cultural distance. As the level of trust deepens, higher-end work can potentially be contracted out because of the increased level of confidence on both sides that work can be carried out effectively at a distance.

In summary, we have noted at least three distinctive aspects of GSAs:

- The manner in which different units of the network are *physically separated and electronically coordinated* across time, space and cultural boundaries.

- The ability to enter into GSAs is no longer restricted to large firms with the inherent capacity to make financial investments, but is also populated with small and innovative ‘born global’ firms driven by technology, ambition, intellectual capital and cost advantages.
- There is a central role for *ICTs*, for coordinating activities across different work units and for defining the content of work. Interdependent work requires the outsourcing and outsourced firms to be linked together by much higher bandwidth than that required for more stand-alone projects. While these *ICTs* help facilitate effective coordination and communication, they come with their own challenges related to access, compatibility, protocols and standards and issues of power and control.

We build on these themes in the next section.

1.3 Nature of GSW

Software development and maintenance activities are the characteristics of processes of the ‘new economy’ involving programmers, software designers and analysts (collectively referred to as ‘knowledge workers’) engaged in designing, developing, testing and implementing software (referred to as ‘knowledge work’). However, the nature of this knowledge is multi-faceted and continuously negotiated and contested by the various actors involved in the software development process (see for example, the case analysis of Sierra in chapter 7). GSW also reflects characteristics of other forms of global work in general where the focus is on developing standardization, productivity and efficiency. Ritzer (1996) labels such work as ‘McDonaldization’. Based on an analysis of fast food restaurants, notably McDonald’s, Ritzer develops a critique of current-day work practices and society as excessively concerned with institutions to *rationalize and control behaviour*. Drawing on Max Weber’s views of rationalization, Ritzer identifies four dimensions of modern institutions:

- Efficiency
- Calculability
- Predictability
- Control.

As the case analysis of Witech in chapter 4 demonstrates, a constant quest in GSAs is to standardize and make efficient various aspects of infrastructure and work practices including, for example, defining the template in which project-related communication takes place. This quest for standardization and efficiency can also be viewed in the historical context of the software engineering tradition, and the continued attempt to impart structure and predictability to software development processes.

GSW involves the application of various kinds of *knowledge systems*, including programming languages, software development methodologies, project management techniques and the application domain. Different programming languages are used in software development, from the older FORTRAN and COBOL to the current Java

and Visual Basic. Several hundred programming languages have been developed for use in both general-purpose and specialized domains. In the 1960s and 1970s, as technology of language compilers developed, large IT firms like IBM, Hewlett Packard and Univac formulated their own languages to support proprietary operating systems and system utilities. Users in other domains also developed their own languages – for example Nortel Networks, a large telecommunications firm, had software for their digital switches written in a proprietary language called Protel. A key technology for GSW came in the 1980s. Common standards increasingly emerged and C and then C++ (considered ‘open’ platforms) became widely used for system software development. Although the development of standards remains contested, developers preferred these open platforms as these did not restrict them to particular technologies, or to specific firms with their proprietary languages and products.

Although global work is not a new phenomenon, distributed software development work is relatively new and begs the empirical question: can approaches to global manufacturing (for example, car assembly plants) or global services (for example, consulting) be transferred seamlessly to software development work? As software work involves physically intangible artefacts whose value is derived from qualities such as efficiency of algorithms, ‘look and feel’ aspects of the user interface, richness of features and so on, this distinction from the production of material goods is useful. Software work has distinctive features, for example, in contrast to manufacturing where production and consumption take place in separate physical domains, services are generally distinguished by the *inseparability* of these functions. This is true of a range of different services from hotels and medical work to legal and accounting practices. However, these services are also starting to be outsourced offshore, as reflected in the growth of firms providing legal and medical transcription services and also those specializing in various transaction-processing functions like billing and ticketing.

Production and consumption are separable to a major degree in software work, where at each stage of the development, artefacts such as program code and documentation enable outputs to be specified and disembedded from the development domain to other use situations. However, information systems research has increasingly established that software design and development is never really ‘finished’, but involves an ongoing interaction and redefinition with the process of use (Bjerknes, Bratteteig and Espeseth 1991). Development and use of software can thus be quite distinct, linked together by various artefacts, and simultaneously be also intricately interconnected. Managing these *complex interdependencies* is a defining aspect of GSW.

Software may be regarded as a knowledge industry but is different from the traditionally accepted knowledge work of consulting in which many aspects rely fundamentally on the expertise of individuals, making it difficult to obtain economies of scale. Software work covers a range of activities including the development of algorithms and user interface designs that require creative talent of the highest order that cannot be scaled up in a mechanical fashion. Friedman (1989) points out that software work of this nature is continually being disciplined, formalized and made subject to managerial control, but

that this is thwarted by factors such as rapid changes in technology and the associated lack of skills in these new domains. Other activities in the spectrum of software work include the work of call centres, data entry and medical and legal transcription that typically need a minimum level of English, typing skills and ability to use a word-processing program. Such work can easily be scaled up with a suitable work place and telecommunications infrastructure, wherever people with these minimal background skills are available in large numbers. In between these extremes, there is a range of activities that demands different degrees of knowledge and skills, and is amenable to varying degrees of scaling up. For example, while maintaining legacy software does not need creative talent of the highest order, it needs individuals who can, in a short span of time, learn new languages, understand the complex relationships in a large piece of software and sensitively operate in the use domain. The extent of separability and scaling, therefore, varies for different software tasks and is significantly shaped by the infrastructure in place, including the available bandwidth, the degree of sophistication of management processes and the prior experience of the partners.

In GSW, tasks at various stages of the software life-cycle may be separated and implemented at different geographic locations coordinated through the use of ICTs. Maintenance and testing were among the first tasks to be outsourced, while early life-cycle tasks such as design and user requirements analysis were considered more difficult to contract out as they required more intimate knowledge of the firm's work practices as compared to maintenance and testing. On the face of it, those types of technology oriented development appear better suited for outsourcing where specifications can be developed and given to an outside party to execute. However, design tasks become harder to undertake because they assume a close familiarity with the market and user preferences. Alternatively, in modular approaches, modules of the software are divided into independent modules and its development 'outsourced' to teams in different locations.

Intangibility, heterogeneity, mobility and scalability are features that differentiate software work from other services and also manufacturing activity. The mental or intellectual activity involved in software work is captured in a form not tangible in the literal sense of being touchable by a human hand but nevertheless is made perceptible through magnetic or optical readers and other devices. The heterogeneity of software work is often limited by the standardization of development processes, methodologies and programming languages. While new and innovative work involves heterogeneity at early stages of conceptualization and design, it requires less at the stages of testing and implementation. Standardization of processes is central to disembedding and fragmenting of software processes to make them amenable to GSW. Perishability, especially important in services like hotels, is not so in software since artefacts like software code and manuals provide mobility with the use of ICTs and enable the life of the software to endure over time.

Another distinctive aspect of software work is the variety of social and human issues that come into play in the phases of design, development, implementation and interpretation of its longer-term implications. Software work, when carried out in a

global setting, magnifies these complexities as it involves relationships of people, teams, organizations and nations with different backgrounds, spoken languages and styles of working in conditions of temporal and spatial separation. Standardization, which is the key to coordinating distributed work, is extremely hard to implement because of the complexities of GSW. Whereas many firms in the manufacturing and services industry try to downplay national and cultural issues through standardization, managers of certain GSAs may capitalize on local idiosyncracies, strengths and creative energies (for example, the case of ComSoft described in chapter 5). While large MNCs are widely seen as weakening of local cultural values, smaller software firms (like ComSoft), in contrast, often attempt to reassert these identities in an effort distinctively to define themselves, drawing upon resources like national and cultural identities in the face of global competition. An ongoing challenge is how to find the appropriate blend between universal solutions and local particularities in a context inherently characterized by a multiplicity of networks.

Another key feature of GSW is the manner and speed in which its knowledge content is subject to change and radical readjustment and is characteristic of work in Castells' (1996) 'network society'. Founded fundamentally on technology and information, GSW involves a new form of 'informational capitalism' in which time, space and knowledge are key resources that entities try to dominate and standardize using various organizational forms. Changes, both technological and organizational, are the norm in GSW. The rapid uptake of Web-based systems by businesses has radically changed the skill sets (like Java) required for software development, for example. Changes are taking place at many levels, from the business models adopted to specific policies and procedures implemented to stem the attrition rate among developers. To deal with these rapid changes, firms need reflexively to monitor and modify their processes on an ongoing basis.

GSW, as we use the term, is broader than the traditional 'software outsourcing' that involved the purchase of goods or services previously obtained internally. Box 1.1 shows Apte's (1990) summary of the activities that were typically outsourced in the past. Prior to the early 1980s, the involvement of foreign companies in outsourced work of this nature was restricted mainly to data-processing or coding-type projects completed by a team of on-site foreign contract programmers. GSW now commonly involves the design and development of new products, support, special services, and whole-life-cycle projects involving different levels of complexity. While 'outsourcing' refers to work contracted out to third-party firms, typically located in the same country, GSW includes work done by subsidiaries and alliances that are necessarily located in a different country. GSW involves turning over to this third party or offshore subsidiary some or all the software development and maintenance tasks, ranging from simple data entry or programming to complete software design, development, data centre operations and full system integration. As the software component in hardware such as telephones, DVDs, cell phones and in cars and airplanes increases, the demand for outsourcing has multiplied (Box 1.1).

Box 1.1 Outsourced IT services

- Data-processing services: data entry, transaction processing, back-office clerical tasks
- Contract programming
- Facilities management: operation and support of data centres
- System integration
- Support operations: maintenance services and data recovery
- Special services: training, hotline support

Source: Apte (1990).

GSAs allow for a range of possibilities both in terms of the kind of projects contracted out and the extent to which the different stages of the development life-cycle can be outsourced. Issues that influence these decisions are the strategic importance of the activity, the degree to which the requirements can be specified and the comparative cost of having the software developed in-house versus having it outsourced. Various arguments have been made for and against companies contracting out core projects: 'never outsource a problem, only a defined task' (Willcocks, Fitzgerald and Lacity 1996; Willcocks and Sauer 2000). Gurbaxani (1996) argues, to the contrary, that innovative projects can be outsourced with support of a strong contracting structure, presence of multiple vendors and a selective outsourcing strategy. In addition to this debate on what should be outsourced, there are also discussions on whether firms should go for 'total' or 'selective' outsourcing, ranging from small stand-alone projects completed through short-term employment of programmers, to projects where the third-party vendor is completely in charge of hardware, software and staff. Total outsourcing can involve design, implementation and maintenance of large projects, or even the support for whole pieces of legacy software or the porting of them to alternative platforms.

While the issues referred to above have been debated extensively in the information systems literature, they remain open empirical questions in the GSW domain. McFarlan (1995) has argued that 'highly structured' projects where processing, file structures and outputs are completely defined, are easy to outsource, as compared to those whose outputs are more open to the users' changing judgement on desirable features, for example, in business process re-engineering projects. Such projects, it is argued, are best done in-house in conditions of co-location and proximity. In addition, to reduce ambiguities in design, Kobitzch, Rombach and Feldmann (2001) argue that structured projects are amenable to an 'engineering' approach which makes it easier to scale up and achieve economies of scale that justify the investments required for establishing a GSW infrastructure. This is related to the maturity and compatibility of the structured processes in both the outsourcing vendor and client organizations. Our empirical work in Japan (see chapter 9) suggests that firms there may be more willing to operate in relatively unstructured projects as compared to North American firms, and this can