Distributed Software Development Based On 24hrkf Of Knowledge Integration And Knowledge Diffusion: Motivation And Recent Developments

Huosong Xia[1], Amar Gupta[2]

[1] Wuhan University of Science & Engineering, P. R. China; [2] University of Arizona, USA

ABSTRACT:

As competition continues to grow through exposure to a global economy, software development will need to adapt to a global mode of operation by facilitating outsourcing and surmounting temporal separations. This paper discusses the procurement and the allocation of optimal global resources for software development via the Global Operations Model (GOM), based on the 24-Hour Knowledge Factory (24HrKF) concept. In order to answer what factors should be considered for global software development, this paper presents a way for distributed software project development Based on 24 hour knowledge factory and GOM.

Keywords: *Knowledge integration, Knowledge diffusion, 24-Hour knowledge factory, Distributed software development*

---

1. Introduction

As our world's competitive global economy continues to develop, software development needs to adapt to a global mode of operation by facilitating outsourcing and surmounting temporal separation. This will require organizational innovation. Specifically, organizational innovation dates back to the work of Joseph Alois Schumpeter, who recognized the importance of technological innovation, and introduced new means of production, new products, and new forms of organization. It is within the framework of innovation theory that we identify the ability to effect the change towards our Global Operations Model GOM model and the 24-Hour Knowledge Factory (Seshasai, Gupta, & Kumar, 2005).

1.1. Problem And Motivation

As IT and global ideas proceed, we must analyze what problems typically arise in software development, as these will presumably be exacerbated when performed across temporal and spatial borders. The following twelve reasons have been cited as factors contributing to software failure (Charette, 2005):

1. Unrealistic or unarticulated project goals

2. Inaccurate estimates of needed resources

3. Badly defined system requirements

4. Poor reporting of the project's status

5. Unmanaged risks

6. Poor communication among customers, developers, and users

7. Use of immature technology

8. Inability to handle the project's complexity

9. Sloppy development practices

10. Poor project management

11. Stakeholder politics

12. Commercial pressures

In this paper, we are mainly concerned with geographically distributed software development issues. The key questions are:

- What characteristics or factors best suit an organization for distributed software development?

- How should we construct the model of global operations for distributed software development?

- How are software development and innovation environment interrelated?

- What factors should be considered for global software development?

Partly in response to the above questions, this paper explores a software development innovation environment system based on the GOM. The paper is organized as follows: Section 2 describes the current state of the global software industry; section 3 analyzes the GOM based on the 24HrKF; section 4 discusses the relationship diagram of enterprise innovation based on the GOM; section 5 discusses the elements and policy of the system for software development in a manner that fosters innovation; and section 6 provides the conclusions.

1.2. Contributions

This paper examines the use of the 24-Hour Knowledge Factory paradigm to effectively manage knowledge flow and work flow in GOM. Our model suggests the use of a relationship diagram of innovation for the distributed software development based on knowledge integration of knowledge space and work space.

2. Related Work

In order to develop a team of professionals ready to effect a 24HrKF scenario, we need to first identify which skills are requisite for this type of dynamic operation. Of the ten most important skills sought in mid-level employees of an IS organization, effective communication is usually deemed to be of the utmost importance (Luftman and Kempaiah, 2007). Harnessing technology, providing business value, managing resources, and executing work are additional IT leadership challenges. The need for these skills is further heightened by the emerging global environment. The skills and technologies utilized by virtual teams are a blend of both old and new. These skills and technologies come with their own advantages and disadvantages (Gillam and Oppenheim, 2006).

A combination of fourteen "new" and traditional skills shows promise for meeting the global sourcing challenges (King, 2007). These fourteen skills are as follows:

1. Contract negotiations and management,
2. Relationship management,
3. Development and implementation of strategic alliances and joint ventures,
4. Vendor and partner assessment and selection,
5. Risk assessment and management,
6. Customization, implementation and integration of a collaborative system,
7. Technology assessment and monitoring,
8. Business Process Redesign,
9. Integrated business and IS planning,
10. Critical systems development and testing,
11. Systems testing,
12. Security,
13. IS personnel development, and
14. Awareness of national cultures.

While current literature on marketing/product management focuses on optimizing revenue, the literature on product line engineering concentrates on lowering costs through reuse (Helferich, Schmid & Herzwur, 2006). A conservative estimate of the annual cost for failed software projects in the US is between $60 billion to $70 billion (Charette, 2005).

We have entered an era of knowledge management, living and working in a knowledge-based society as knowledge workers. Various streams of Knowledge Management (KM) research have gradually emerged. Early research focused on understanding the differences between data, information, and knowledge classifications, such as tacit and explicit knowledge (Nonaka and Takeuchi, 1995), and individual and collective knowledge (Spender, 1996). Other research viewed knowledge as a source of competence and as a competitive resource (Hung et al., 2001). A goal of many KM initiatives is to develop a global knowledge community where knowledge is shared and utilized by various practicing organizations in the community. However, knowledge sharing is difficult and only partial knowledge can be shared and created between parties. The use of information technology to support knowledge sharing within and between communities of practice has been explored by many researchers (Pan and Leidner, 2003). The advent of different knowledge management techniques has transformed decision making processes. In order to address the problem of knowledge sharing, this paper describes a software development innovation environment that holds representations of descriptive, procedural, and reasoning knowledge.

A report of the Association for Computing Machinery (ACM) delineates six varieties of work related to IS that are often offshored: (1) programming, software testing, and software maintenance; (2) IT research and development; (3) high-end jobs, such as software architecture, product design, project management, IT consulting, and business strategy; (4) physical product manufacturing; (5) business process outsourcing/IT Enabled Services; (6) call centers and telemarketing (Aspray, Mayadas & Vardi, 2006). Additionally, there are three phases of socialization in global distributed teams: creation, maintenance, and renewal. Harnessing socializing processes to support globally distributed collaboration is not easy (Oshri, Kotlarsky & Willcocks, 2007). Traditional software development is based on the closed-world assumption that the boundary between systems and environment is known and unchanging (Baresi, Nitto & Gbezzi, 2006). However, today's unpredictable open-world settings demand techniques that will allow software to react to changes by being self-organizing its structure and self-adapting its behavior.

The global operations model (GOM) is a key driver for global sourcing in software development. The software industry is different from, say, the traditional automobile industry where Taylorism and Fordism (Doray, Macey & Godelier, 1988) have led to the standardization of manufacturing and management. With Tayloristic processes, product diversification is limited but productivity is improved due to the specialization of workers performing specific small tasks. The tasks of software development can be decomposed into clearly defined processes, including requirements analysis and specification, top level design and specification, detailed design, implementation, testing, and service. This decomposition allows for easier outsourcing in networked and virtual organizations.

The concept of the "24-Hour Knowledge Factory" (24HrKF) is an idea that faciltates global software development (Seshasai, Gupta & Kumar, 2005). In this case, work continues around the clock; each member of the team works during normal workday hours specific to his/her geographic location. The 24HrKF concept offers the following potential benefits:

- Reduces time and costs,
- Facilitates use of low-cost labor and production,

- Improves the quality and efficiency of customer service,
- Increases flexibility,
- Helps manage growth and value creation (Seshasai, Gupta & Kumar, 2005, Gupta, Seshasai, Mukherji, Ganguly, 2007).

Nonaka advises managers of international businesses to build their organizations around the core process of creating knowledge (Nonaka 1991). Specifically, this was the key to the success of Japanese enterprises that have learned how to transform tacit knowledge into explicit knowledge. Nonaka proposed mechanisms through which enterprises could design their organization and define managerial roles and responsibilities. This is the "how" of the knowledge-creating enterprise: the structure and practice that translates an enterprise's vision into innovative technologies and products. Spangler and Peters (2001) examined the distributed knowledge model and actions in complex systems, analyzing the implications of decision support mode and the structure of relationships in complex systems development. Bloodgood and Salisbury (2001) highlighted the influence of organization change strategies on information technology and knowledge management strategies, and analyzed the relationship between knowledge innovation, knowledge transformation, knowledge management strategies, and organization strategies.

Enterprise knowledge management is aimed at strengthening knowledge sharing. Dixon (Dixon, 2000) identified the following five different kinds of knowledge transfer processes:

1. *Serial transfer* means that the knowledge a team has learned from doing its task can be transferred and used the next time that the team does the task in a different setting.
2. *Near transfer* means that the explicit knowledge a team has gained from doing a frequent and repeated task that the organization would like to replicate in other teams that are doing very similar work.
3. *Far transfer* means that the tacit knowledge a team has gained from doing a non-routine task that the organization would like to make available to other teams that are doing similar work in another part of the organization.
4. *Strategic transfer* means that the collective knowledge of the organization needed to accomplish a strategic task that occurs infrequently but is of critical importance to the whole organization.

1. *Expert transfer* means that the technical knowledge a team needs that is beyond the scope of its own knowledge but can be found in the special expertise of others in the organization (Dixon, 2000).

The guidelines for designing each of the aforementioned types of transfer processes ensure that the system is effective in identifying the owner manager, employment manager, hierarchical structure, international matrix, combined enterprise, network structure and alterable network structure and team through the evolvement of enterprises' organizational structures (Dixon, 2000). The evolution of the organizational structure requires knowledge sharing to ensure the match between knowledge transfer and different types of knowledge in an organization. Dixon, undertook an in-depth study of several American organizations that are leading the field in successful knowledge transfer.

3. GOM Based On 24hrkf Concept

Depending on its place in the software industry chain, each country or zone possesses specific advantages. Synergetic and global operations have had major impact on software development. Friedman (2005) distinguishes between three great eras. The first great era, Globalization 1.0, lasted from 1492 to around 1800, when Columbus set sail, opening trade. It shrank the world from 'large' to 'medium'. The second great era, Globalization 2.0, lasted roughly from 1800 to 2000. This era shrank the world from a 'medium' to 'small'. Around the year 2000, when we all entered what Friedman calls Globalization 3.0, the world was completely flattened, leading to a new need for individuals and companies to collaborate and compete globally (Friedman, 2005). In Globalization 3.0, the process of global software development involves making changes to the structure of organization and the creation of new information infrastructure. In particular, new mechanisms are needed for organizing flows of work and knowledge in geographically distributed settings.

The key components of knowledge, in the context of software development, are described in figure 1. Critical success factors include consideration of time zones, demand management, long-term productivity, integrated value chains, organizational models, barriers within firms, and location choices (Gupta, et al, 2007).

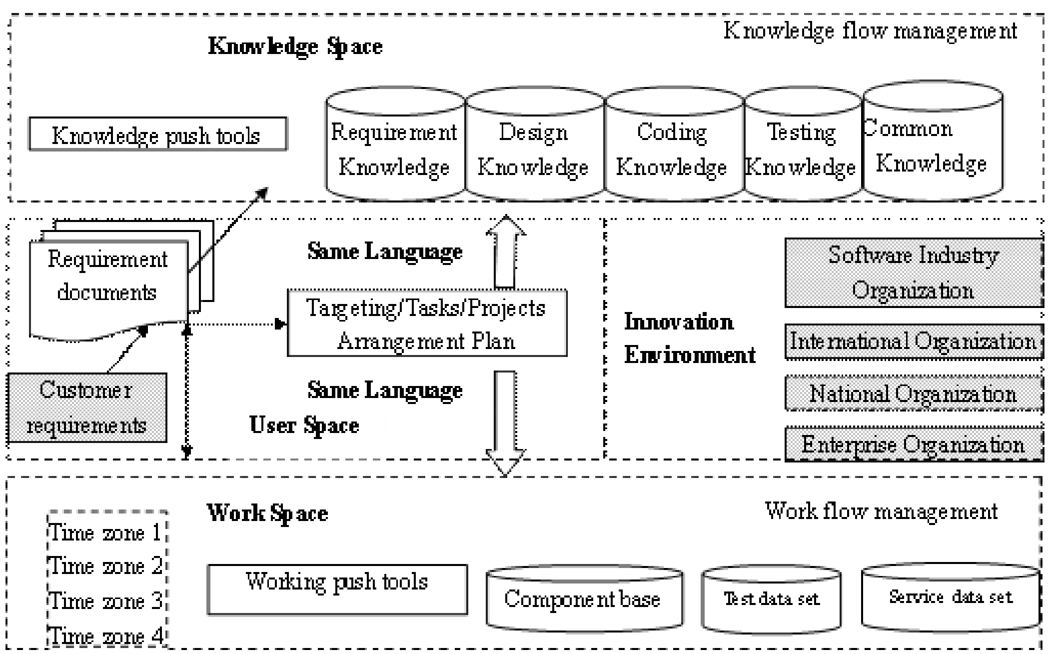

Figure1 Knowledge Space and Work Space of Software Development

In particular, efficient work flow management and knowledge flow management are important aspects. Since the work space involves a global space with different time zones, the component base, the test data set, and the service data set must be effectively shared within the boundaries of the global software endeavor. Working push tools are the tools that can facilitate the task from the upper unit (late hours) in one time zone to the low unit (early hours) in another time zone. The transition can be facilitated using the Signboard Production Control System, similar to the one introduced by Toyota, in the 1950's, in assembly line operations. In Toyota's dual-card signboard system, there are two main types of signboards: a production signboard, which signals the need to produce more parts; and a withdrawal signboard (also called a "move" or a "conveyance" signboard), which signals the need to withdraw parts from one work center and deliver them to the next work center (http://personal.ashland.edu/~rjacobs/m503jit.html).

With respect to knowledge flow management, five types of knowledge exist: requirements knowledge, design knowledge, coding knowledge, test knowledge, and common knowledge. Some knowledge is static and easily coded for storing in the knowledge base. Other knowledge is dynamic and needs to be communicated in new ways in order to surmount the barriers of spatial and temporal separations. Knowledge push tools are similar to working push tools. However, the knowledge tools are more complex, and require more study.

Table 1: Tools For Development Of Software

Type	Tools	Coding Environment	Characters
B/S	ASP:.NET, ASP	Text compile, Frontpage, Dreamweaver and IDE by software itself	Easy, quick, medium security, transferring to different platform is not easy
	JSP: Java, JSP	Text compile, Frontpage, Dreamweaver and IDE by software itself	Difficult to grasp, coding is slow, high degree of security, transfering to different platform is easy, many free tools are available; open source coding
	PHP: php	Text compile, Frontpage, Dreamweaver and IDE by software itself	Easy, quick, medium security, transferring to different platform is slightly difficult, many free tools and open source coding
C/S	PB	IDE by software itself	Easy, quick, good matched database, high security, transferring to different platform is easy
	Delphi	IDE by software itself	Easy, quick, poorly matched database, high security, transferring to different platform is easy
Mixed	VB VC++, C	IDE by software itself	Slow in terms of development, runs fast, poorly matched database, high degree of security, transferring to different platform is not easy

Design knowledge is one of the five types of knowledge space. One type of knowledge necessary for design is incorporated within the tools of development software. Tools for development of software in software design knowledge are summarized as shown in Table 1.

The 24HrKF concept is with a special case of globally distributed software development (Seshasai, Gupta & Kumar, 2005, Gupta, et al, 2007). The global operation model for software development based on the 24-Hour Knowledge Factory is presented in Figure 2. As an extension to this model, overlap of work between neighboring time zones can help to ensure better communication with the concerned teams.

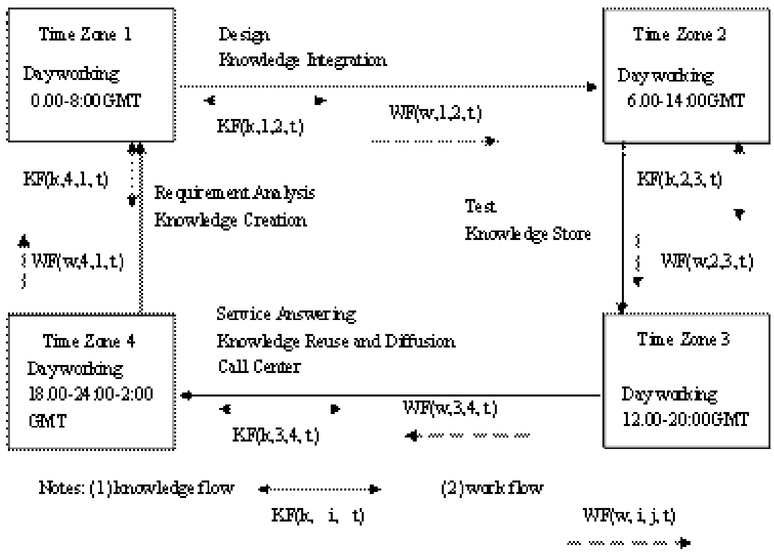

Fig. 2 Global Operation Model Based on 24 hour knowledge factory

Knowledge can be transferred across distributed teams using an array of communication tools, variations of case based reasoning techniques, and knowledge sharing tools such as collaborative software, Web-DAV, extreme programming, blogs, and Wikis (Gupta and Seshasai, 2007).

The software development cycle involves multiple phases for handling requirement analysis activities, design activities, testing activities, and service activities respectively. Distributed software project development based on 24 hour knowledge factory and GOM is presented as Fig. 3

These activities can be performed in different time zones. We formulate the following model:

Let

- SP={RA,DA,TA,SA} be the set of activities in software project development,
- RA=$\{r_{a1},r_{a2}\ldots r_{an}\}$ be the set of requirement activities,
- DA=$\{d_{a1},d_{a2}...d_{am}\}$ be the set of design activities,
- TA=$\{t_{a1},t_{a2}\ldots t_{al}\}$ be the set of test activities,
- SA=$\{s_{a1},s_{a2}\ldots s_{ar}\}$ be the set of service activities.

- KF (*k*, *i*, *j*, *t*) be the knowledge flow *k* of the node *i* to *j* at time *t*.

The common knowledge flow k can be expressed by *i*=0, *j*=0, *t*=0. The common knowledge flows are stored in static knowledge bases, and the dynamic knowledge flows are stored in push pool knowledge systems.

Let WF (*w*, *i*, *j*, *t*) be the work flow w of the node *i* to *j* at time *t*. The dynamic work flows are stored in push tools of a work flow system similar to that in the card signboard system.

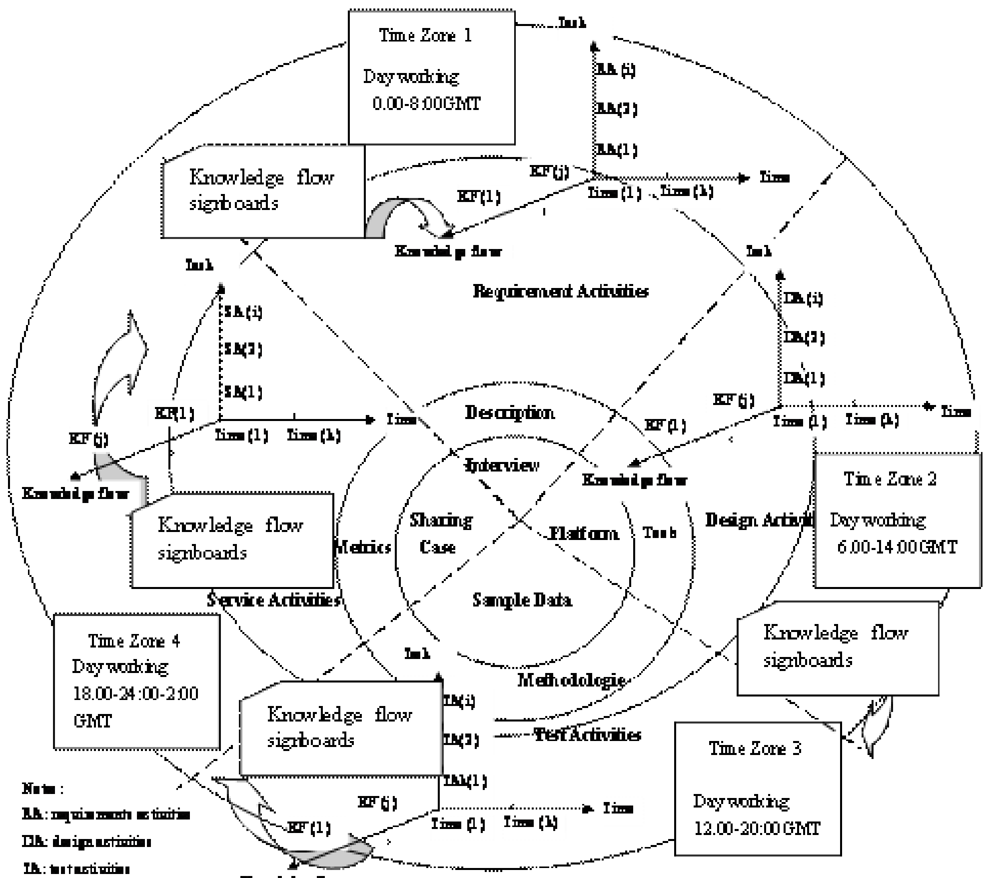

Fig. 3 Distributed software project development Based on 24 hour knowledge factory and GOM

There are two main types of signboards in our GOM: knowledge flow signboards, which signal the need for knowledge reuse and sharing; and work flow signboards, which signal the need to transfer components from one team in a time zone and deliver them to the next team in the next time zone.

In order to encourage team members in different time zones to contribute their knowledge and ensure high quality of processes, we need to build a performance model for software development. Each activity has involves knowledge usage or creation, and has its own characteristic metrics. We describe the relationship of the activities, work flow, and knowledge flow in Table 2.

Table 2: The Relationship Of The Activities, Work Flow, Knowledge Flow

Phases	Activities	Work Task	Knowledge Diffusion	Metrics	Delivered
Phase 1	$r_{a1}, r_{a2} \ldots r_{an}$	$WF_{r1} \ldots WF_{rn}$	$KF_{1} \ldots KF_{rm}$	$M_{r1} \ldots M_{rn}$	WF (*w*, 1, 2, *t*), KF (*k*, 1, 2, *t*)
Phase 2	$d_{a1}, d_{a2} \ldots d_{am}$	$WF_{d1} \ldots WF_{m}$	$KF_{d1} \ldots KF_{dm}$	$M_{d1} .. M_{dm}$	WF (*w*, 2, 3, *t*), KF (*k*, 2, 3, *t*)
Phase 3	$t_{a1}, t_{a2} \ldots t_{al}$	$WF_{t1} \ldots WF_{tl}$	$KF_{t1} \ldots KF_{tl}$	$M_{t1} .. M_{tl}$	WF (*w*, 3, 4, *t*), KF (*k*, 3, 4, *t*)
Phase 4	$s_{a1}, s_{a2} \ldots s_{ar}$	$WF_{s1} \ldots WF_{sr}$	$KF_{s1} \ldots KF_{sl}$	$M_{s1} .. M_{sr}$	WF (*w*, 4, 1, *t*), KF (*k*, 4,1, *t*)

The metrics in phase 1 including $M_{r1} \ldots M_{rn}$ are assigned by customers and designers. The metrics in phase 2 including $M_{d1} \ldots M_{dm}$ are determined by standard rules; they are assigned by testing members in phase 3. The metrics in phase 3 include $M_{t1} \ldots M_{tl}$ that are given by users and maintainers.

Therefore, the GOM based on the 24HKF concepts can potentially serve as the basis for software development that transcends geographic boundaries and spans different time zones for knowledge and work flows.

4. Recent Developments: Software Industry Categorization

The first countries to develop software industries primarily for export, rather than domestic purposes, were Ireland and Israel (Cusumano A. M., 2005a). India's software export industry began in 1974 and has seen rapid growth since the late 1990s. The United States has historically dominated and continues to dominate the software and services industry, accounting for 80% of global revenue (Cusumano A. M., 2005a). Of the roughly $285 billion in total revenues of the global industry in 2004, only about $80 billion was generated by non-US companies (Aspray, Mayadas & Vardi, 2006). The U.S. and Canada account for about half of the worldwide software business, and Europe accounts for approximately 30% (Cusumano A. M., 2005a). Asia accounts for 15% to 20%, with Japan comprising about 10% of the world market and representing the single largest Asian market (Cusumano A. M., 2005a).

The software industry can be divided into three streams of innovation and characters: lower, middle, and upper streams. Each stream has particular characteristics, a distinct value-added model, and typically occurs in particular countries or zones.

The United States is in the upper stream of the software industry chain, with control of prominent operating systems, Database Management Systems (DBMS), and Web interface platforms. The upper stream has the key characteristics of high value-addition through unique intellectual property and system software. The business model of the upper stream includes a strong innovation environment and software industry leadership. Due to their global source cost advantage, companies in the upper stream are well-positioned for global operations through offshoring to the middle and low stream.

The software industries in Europe, Japan, Ireland and India operate in the middle stream of the global software industry, where the focus is on middleware, embedded software, and module development (Aspray, Mayadas & Vardi, 2006). The middle stream software industry sector can extend to the upper stream, as well as outsource to the lower stream. In the Japanese software industry, for example, Hitachi, Fujitsu, NEC, and Toshiba established software factories, successfully built large-scale systems, and customized industrial applications relying heavily on standardized development processes, rigorous quality-assurance techniques, extensive tool support, and reuse libraries (Cusumano A. M., 2005a). Excluding billions of dollars worth of "hard" products that contain embedded software, ranging from machine tools to consumer electronics and automobiles, Japanese firms have made relatively few software products during the past two decades (Cusumano A. M., 2005b). The Japanese software industry remains far behind the software industry in the west, and software development in Japan remains costly. In contrast, Indian companies have benefited from ubiquitous English language skills, much lower wage levels, a strong process focus, and excellent university training in computer science, mathematics, and engineering subjects (Cusumano A. M., 2005a, 2005b, 2006).

China is an example of of a country in the lower stream of the global software industry, focusing specifically on applications software. The business model of the lower stream is characterized by a weak innovation environment and low addition in value. The total revenues from software sales in China were about $60 billion (RMB 4,80 billion) in 2006 (http://www.sina.com.cn, 2007-01-02 news). There are 12,400 software companies, with total sales revenues of the ten biggest software companies in China being below ten billion dollars. The total revenues from software outsourcing in China is $1.4 billion (http://www.csia.org.cn/, http://www.chinabpo.org.cn/), with the largest company in the arena, NeuSoft Group Ltd in China, accounting over $100 million in 2006 (http://www.csia.org.cn/, http://www.chinabpo.org.cn/). Figure 4 summarizes the different categories of software and the current state of affairs.

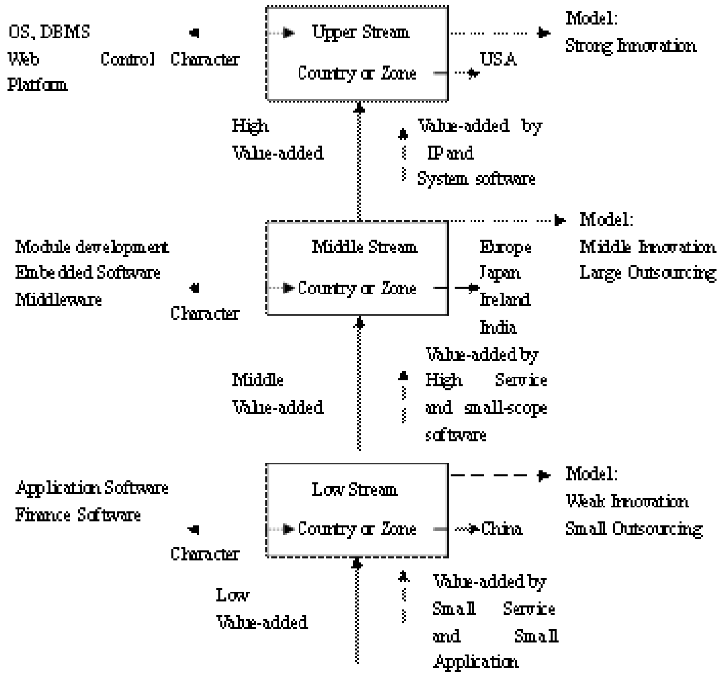

Fig. 4 Software Industry Chain

5. A Relationship Diagram Of Software Industry Innovation Based On GOM

The forces that motivate and impact innovation in the context of the software industry in a global economy are graphically depicted in Figure 5. Innovations in the area of organization structures can help to optimize operations.

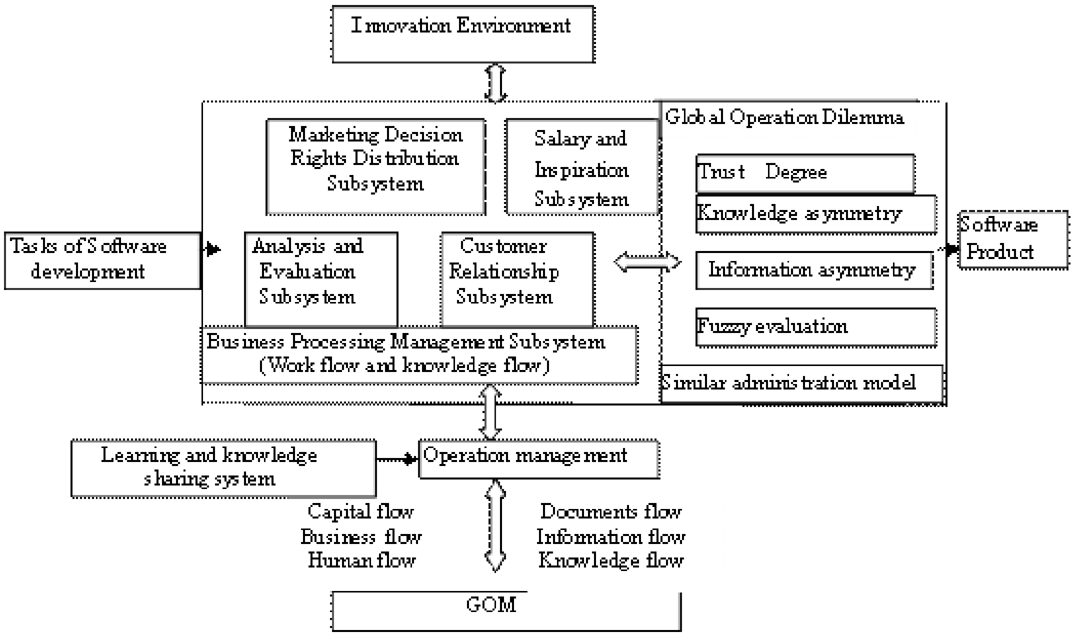

Fig. 5 A relationship diagram of software organization innovation based on GOM

6. Connclusions

Distributed software development requires virtually space across geographic and temporal boundaries based on knowledge integration and knowledge diffusion because of the change of GOM. According to its position in the global software industry, we may select different model of distributed development. Distributed software project development based on 24 hour knowledge factory and GOM is good model. The model can maintain sustainable competitive advantage for different countries or zones in the chain of software industry.

The aims of studying the chain of software industry and distributed software are to improve software development in a global operation perspective and to find a relationship of software development in the innovation environment. For these aims, this paper not only discusses a scenario of GOM based on the 24HrKF concept, but also summarizes recent developments: categorization of software industry. From an analytic, we belive that distributed software development needs to be created a relationship diagram of software organization innovation based on GOM. From knowledge management, the model of knowledge flows and work flows can improve software development based knowledge integration and knowledge diffusion. From business viewpoint, the concept of 24HrKF concept is relevant much factors and innovation environment based on GOM, the concept is not only benefit for distribution software development in innovation environment, but also become "multi-win" situation for all software organization in different country and different time zones.

7. Acknowledgements

This research was supported by National Philosophy and Social Science Fund (07BTQ010) of China, HuBei Province Fund of China (2007097, 2008d062, 2008244)), CNTAC Fund (2007082). We appreciate the help provided by Yang Qian, Igor Crk, Sarah Trainor and David Branson Smith.

8. References

Aspray W., Mayadas, F.. and Vardi, M.Y. (2006). "Globalization and offshoring of software: A report of the ACM job migration task force." An Association for Computing Machinery Report.

Aspremont, C., Bhattacharya, S and Gerard-Varet L. (1998). "Knowledge as a public good: efficient sharing and incentives for development effort." Journal of Mathematical economics. 30, 389-404.

Baresi, L., Nitto, Di E.. and Gbezzi C. (2006). "Toward Open-world software: issues and challenges." Computer Magazine. 10, 36-43

Bloodgood, M. J. and Salisbury, W. D. (2001). "Understanding the influence of organizational change strategies on information technology and knowledge management strategies." Decision Support Systems. 31, 55-69.

Charette, R.N. (2005). "Why software fails (software failure)." Spectrum IEEE 42(9), 42-49

Cusumano, A. M. (2006). "Where does Russia fit into the global software industry?" Communications of the ACM No. 49 (2), 31-34.

Cusumano, A. M. (2005a). "Software in Ireland: A balance of entrepreneurship and lifestyle management." Communications of the ACM 48(10), 25-27

Cusumano, A. M. (2005b). “The puzzle of Japanese software.” Communications of the ACM No. 48 (7), 25-27.

Dixon, M. N. (2000). “Common knowledge: how companies thrive by sharing what they know.” Boston, Mass: Harvard business school press.

Doray,B., Macey, D. and Godelier,M.(1988). From Taylorism to Fordism: A Rational Madness Free Association (London).

Friedman, L. T. (2005). “The world is flat--a brief history of the twenty-first century.” Farrar, Straus and Giroux (New York).

Gillam, C. and Oppenheim, C. (2006). “Review articles: Reviewing the impact of virtual teams in the information age.” Journal of information science. 32(2), 160-175.

Gupta, A. (2002). “The Critical Role of Trusted Agents.” Sloan Management Review. 4

Gupta, A. and Seshasai, S.(2007). “24-Hour Knowledge Factory: using Internet Technology to leverage spatial and temporal separations.” ACM Transactions on Internet Technology. 7 (3), 1-22.

Gupta, A., Seshasai, S., Mukherji, S., Ganguly, A. (2007). “Offshoring: The transition from economic drivers toward strategic global partnership and 24-Hour Knowledge Factory.” Journal of electronic commerce in organizations, 5 (2), 1-23.

Helferich, A., Schmid, K. and Herzwur G. (2006). “Product management for software product lines: An unsolved problem.” Communications of the ACM. 49 (12), 66-67.

Holsapple, C. W. and Joshi, K. D. (2001). « Organization knowledge resources.” Decision Support Systems. 31, 39-54.

Hoving, R. (2007). “Information technology leadership challenges-past, present and future.” Information systems management. 24, 147-153. rsma, W. (2005). “Que

Hung, J, C, M., Newell, S., Pan, S.L., (2001).The process of global knowledge integration: an empirical study of a multinational investment bank’s Y2K program. European Journal of information systems. 10(3),161-174.

Iske, P. and Boe stion and answer systems for knowledge sharing: concepts, implementation and return on investment.” Journal of Knowledge Management. 9 (1).

Jalote, P. and Jain, G. (2006). “Assigning tasks in a 24-hour software development model.” The Journal of Systems and Software. 79, 904–911, 905.

King, R. W. (2007). “The IS organization of the future: Impacts of global sourcing.” Information Systems Management. 24, 121-127.

Lagerstrom, K. and Anderson, M. (2003). “Creating and sharing Knowledge within a transnational team-the development of a global business system.” Journal of world business. 38, 84-95.

Luftman, J. and Kempaiah, M. R. (2007). “The IS organization of the future: the IT talent challenge.” Information system management. 24, 129-138.

Nonaka, I. (1991). “The knowledge creating company.” Harvard Business Review, 96-104.

Nonaka, I. and Takeuchi, H. (1995). The knowledge-creating company. Oxford University Press.

Oshri, I., Kotlarsky, J. and Willcocks, L. (2007). “Global software development: Exploring socialization and face-to-face meetings in distributed strategic projects.” Journal of strategic information systems 16, 25-49.

Pan, L. S. and Leidner, E. D. (2003). “Bridging communities of practice with information technology in pursuit of global Knowledge sharing.” Strategic Information Systems 12, 71-88

Sakakibara, M. (2003). “Knowledge Sharing in Cooperative research and development. Managerial and Decision Economics.” 24, 117-132.

Seshasai, S., Gupta, A., and Kumar, A. (2005). “An integrated and collaborative framework for business design: A knowledge engineering approach.” Data & Knowledge Engineering. 52, 157–179.

Soekijad, M. and Andriessen, E. (2003). “Conditions for Knowledge sharing in Competitive Alliances.” European Management Journal. 21,578-587.

Spangler, E. W. and Peters, M. J. (2001). “A model of distributed knowledge and action in complex systems.” Decision Support Systems. 31, 103-125.

Spender,J. C.(1996).Organizational knowledge,learning and memory:three concepts in search of a theory. Journal of Organizational Change Management,9(1),63-68.

Xia, H. S., Du, K. Q. and Cai, S. Q. (2003). « Enterprise knowledge tree model and factors of KMS based on E-C.” Journal of Knowledge Management. 7 (1).

Zhuge, H. (2002). “A Knowledge grid model and platform for global knowledge sharing.” Expert Systems with applications. 22, 313-320.

About the Authors:

Huosong Xia is a professor in the Department of information management and information systems of the Wuhan University of Science and Engineering. He was a visiting scholar at Eller College of Management of the University of Arizona, USA in 2005. His main research interests are: data mining and knowledge management, MIS, DSS, e-Commerce and Logistics Information System. His publications have appeared in refereed international journals such as *Journal of Knowledge Management, International Journal of Knowledge Management, Journal of Systems Science and Information,* *Journal of Grey System*. He may be contacted at E-mail Box: bxxhs@sina.com; xiahuosong8@gmail.com

Amar Gupta is Thomas R. Brown endowed professor of Management and Technology; Professor of Entrepreneurship, Management and Organizations, and MIS; Senior Director for Research and Business Development; Eller College of Management (Also Professor of Computer Science, College of Science), at the University of Arizona, USA. Earlier, he was at MIT from 1979 to 2004 and served as the founding Co-Director of the Productivity from Information Technology (PROFIT) Initiative. His current areas of interests include: Knowledge Acquisition; Knowledge Discovery and Data Mining; Knowledge Management; and Knowledge Dissemination. He is an Associate Editor of ACM Transactions of Information Technology (TOIT)