Knowledge Management in Global Software Teams

Sangeeta Shah Bharadwaj and Kul Bhushan C Saxena

Information technology (IT) organizations, especially software development organizations, are knowledge-intensive firms where the knowledge is mainly embedded in human beings and is largely in the form of tacit knowledge. Managing knowledge in global software teams is very critical as knowledge is a source of competitive advantage for these organizations. They have adopted emergent team-based structures as a response to changing business needs and are globally distributed. Sharing of tacit knowledge requires more people-to-people interaction which is impossible in these organizations. Due to this reason, it is essential to manage certain critical knowledge during the progress of the projects related to achieving the performance goals and the learning goals to consistently sustain and improve project performance. This study identifies the following critical knowledge areas related to the learning goals:

- user requirements knowledge
- functional domain knowledge
- technical knowledge
- project status knowledge
- project experience knowledge.

A five-layered knowledge management framework has been applied to model the software team knowledge. This model is suggested as a process approach to team knowledge management to strengthen knowledge management in software teams. As per the knowledge management framework, all the identified knowledge related to the project are not well managed. One of the reasons for not managing well a particular type of knowledge is the absence of knowledge management processes. The global software teams share knowledge through a virtual space as against a real physical platform with proper IT infrastructure in place. Due to the distributed nature of the teams, rules, conventions, and sharing of norms is already put in place. It, thus, helps in managing project status knowledge, domain knowledge, and technical knowledge. It also promotes management of requirements knowledge and project experience knowledge. However, only ad hoc processes which are immature are in place to manage the knowledge areas. The tools of team knowledge management and leadership commitment are the next two layers of the model to manage the software team knowledge.

This study summarizes the status of the following critical knowledge areas related to the learning goals:

- The most critical knowledge area is the user requirement knowledge. Though newer processes are introduced to manage the same, managing user requirements still remains a challenge for the members of the global software teams.
- Functional domain knowledge and technical knowledge are managed well by companies but technology updates have put pressure in identifying the gaps and bridging it during the project execution.
- Project status knowledge has been well managed in the global software teams with the help of formal procedures and documentation. The Capability Maturity Model (CMM) certification requirement of IT organizations is facilitating this knowledge management area.
- Capturing and reusing the project experience knowledge of the existing projects and clients is still an open issue.

The layered knowledge management framework will help in implementing knowledge management processes for each critical knowledge area.
In the present global and hyper-competitive business environment, organizations require fast response to the changing business needs. Businesses, especially software development organizations, are knowledge-intensive and knowledge is the only input that can help these organizations to cope with radical changes. In the software industry, knowledge is a strategic resource, i.e., if knowledge is used strategically, it could enhance the quality of developed software at less cost (Prokesch, 1997). These organizations engage in the development of large and complex software systems which are also characterized by uncertainty (due to intangibility of software, application of new technology, and difficulty in eliciting user requirements) and the need for extreme precision (numerous lines of code need to interact appropriately to ensure reliable operations of eventual system). A huge amount of knowledge is created during the life-cycle of these projects which the organizations need to capture. But, though there are processes to capture the knowledge created through experiences in one project, they are not shared or applied to the next project in an effective manner. To add to this is the high turnover of people causing loss of experiential knowledge. Hence, there is a need for knowledge management in these organizations.

The IT organizations have adopted team-based structures. To make situations even more complex, often, these teams are geographically distributed. Tacit knowledge such as intuition — the expertise which requires people-to-people contact for sharing the knowledge — is impossible in such teams. The members of such a globally distributed team interact and collaborate through information and communication technology (ICT)-based systems. This ability gives such teams increased flexibility and responsiveness permitting them to rapidly form a group of dispersed and disparate experts into a virtual team that can work on projects. When finished, the team can be disbanded and members redeployed to other projects. However, in the absence of face-to-face interaction, sharing knowledge that is tacit form does not take place. Only the explicit knowledge that is well-documented can be easily and effectively shared among the team members.

Thus, knowledge management in global software teams is one of the challenges for the organizations in the software industry. The members of global software teams require sophisticated knowledge and their expertise is a source of competitive advantage for the organization (Ekstedt, 1989; Winch and Schneider, 1993).

According to Katzenbach and Smith (1993), for a global software team to become a successful high-performance team, one of the key characteristics is that the team members communicate and learn whatever is necessary to get their job done. In other words, fast learning and open communication are two important pre-requisites. But, they are probably not sufficient. For knowledge-intensive team tasks, such as software development, a systematic contribution of individual expertise is a crucial element. Thus, knowledge at the individual level of the team members needs to be institutionalized and localized at the organizational level in the form of collective frames of references, systematized methods of work, and sophisticated routines and processes (Starbuck, 1992; Alvesson, 1995; Morris and Empson, 1998). Hence, a comprehensive model of team knowledge management has to be developed. It is in this context that we discuss the critical knowledge management areas of global software teams and a framework to manage the same.

GLOBAL SOFTWARE TEAMS: A BACKGROUND

In the current knowledge-based economy, companies are increasingly adapting complex IT-based systems to support their core business and management processes. Consequently, more than a decade ago, many companies began experimenting with outsourcing from remotely located software development facilities to seek access to lower costs and skilled resources (Sahay, Nicholson and Krishna, 2003). As a result, software development has increasingly become a multi-site, multi-cultural, globally distributed undertaking. Essentially, software development has been described as a complex problem-solving process simultaneously involving a number of individuals, teams, and organizations with competing goals, interests, and responsibilities (Carmel, 1999). Multi-site software development can take several different forms such as the following (Kobitzsch, Rombach and Feldman, 2001):

- Separate teams in basically independent companies, e.g., two technical teams, one working in vendor organization and the other working in client organization in different countries and related to each other with respect to the common software project the teams are working for.
- Separate teams in legally related companies, i.e., two
teams of legally related companies working in two different countries, e.g., the off-shore development centres (ODC) of client organizations in India.

- One team distributed across multiple sites of legally related companies, e.g., the project teams of a vendor organization situated at more than one location in different countries.
- One team distributed across multiple sites of several basically independent companies as is mostly the case with globally distributed software teams where the vendor has teams at the development centre, let us say in India, a team in the country where the client is, and the client team in the same country.

**KNOWLEDGE MANAGEMENT AND ORGANIZATIONAL KNOWLEDGE**

In the most basic form, knowledge can be thought of as an information that is ‘contextual, relevant, and actionable’ (Soliman and Youssef, 2003; Wainwright, 2001). Knowledge management can be defined as a systematic discipline and a set of approaches to enable information and knowledge to grow, flow, and create value in an organization. This involves people, information, workflows, best practices, alliances, and communities of practice. Knowledge management is also broadly understood as any process or practice of creating, acquiring, capturing, sharing, and using knowledge wherever it resides to enhance learning and performance in organizations (Quintas, Lefrere and Jones, 1997) (Figure 1). It encompasses both types of knowledge — tacit and explicit. Explicit knowledge is knowledge that can be codified. When a company’s employees rely on explicit knowledge to do their work, the people to document approach makes the most sense. Tacit knowledge, by contrast, is difficult to articulate in writing and is acquired through personal experience, i.e., a people-to-people approach. Knowledge management strives to cut across organizational boundaries, enhance communication and collaboration, and transform local knowledge into organizational knowledge (Duffy, 2000; Lee and Hong, 2002).

Knowledge management needs the right methods, technologies, and tools for successful implementation (Duarte and Snyder, 1999, Duffy, 2001; Gupta, Iyer and Aranson, 2000; Marwick, 2001). A knowledge management system facilitates knowledge sharing ensuring knowledge flow from the person(s) who know(s) to the person(s) who need(s) to know throughout the organization while knowledge evolves and grows during the process. Organizations need to give incentives to its people for sharing knowledge (Hansen, Nohria and Tierney, 1999).

The acquisition, interpretation, and practice of knowledge by an individual or an organization is termed as learning, a step further to knowledge. Organizational learning refers to the development of insight and association of past actions, the effectiveness of those actions, and future actions. The capability of an organization to adapt to the changing environment (Hedberg, 1981; Nanda, 1996) refers to organizational knowledge as a step further to organizational learning. Organizational knowledge is the resource generated through accumulation of organization learning. It is internally generated and evolves through the process of learning.

**KNOWLEDGE MANAGEMENT IN GLOBAL SOFTWARE TEAMS**

The different groups of clients, programmers, designers, testers, and project managers of a global software team are grouped in the form of project teams. These project teams are facing a continuously increasing demand for quality improvement in their products and services to compete in the competitive market. In order to survive, they need to improve their knowledge faster than their competitors. Law and Chuah (2004) talk about a project-team-based learning framework to promote knowledge creation and sharing. The software project teams are struggling to manage well the nine knowledge areas:

![Figure 1: Knowledge Management Life-cycle](image-url)
cost, time, scope, risk, communication, team members, procurement, quality, and integration related to project management knowledge areas in the project management body of knowledge. However, it is quite difficult for the project teams to learn from project experiences. The nine knowledge areas can be related to the performance goals for successfully completing the project. The other goal of the project team, i.e., the learning goal, largely remains undefined. In this paper, knowledge areas related to the learning goals have been identified as user requirement knowledge, domain knowledge, technical knowledge, project status knowledge, and experiential knowledge typical to software projects. Project performance can be improved with time if knowledge areas related to the learning goals are managed well. Waterson, Clegg and Axtell (1997) have emphasized that software development involves a variety of issues concerning the communication and coordination of knowledge relating to the programme, the methodologies to be used, the domain area, and various organizational practices such as reporting relationships within the project team. Managing these processes by which knowledge is acquired, shared, and integrated between these various individuals, teams, and organizations is a crucial task in the process of software development (Walz, Elam and Curtis, 1993).

The members of globally distributed teams do not have all the knowledge required for the project and they need to acquire additional information and knowledge from different sources such as relevant documentation, formal training sessions, results of trial and error exercises, and other team members (Walz, Elam and Curtis, 1993). Several authors have emphasized how routines and standard operating procedures are used to transfer knowledge (Cohen and Bacdayan, 1994; Huber, 1991; Levitt and March, 1988). The process of acquisition and sharing of knowledge of global teams is problematic as most of the knowledge is tacit knowledge and is embedded at the individual level as these are knowledge-intensive firms. The process is enabled by information technology features, established sharing practices, and norms within teams (Majchrzak et al., 2000; Malhotra et al., 2001). Software development is no longer a homogeneous field. A socio-technical approach and a commitment to project management principles are essential for attaining success in software development projects. But, managing project knowledge is another critical factor that has to be taken into consideration. Managing knowledge in globally distributed teams involves managing software projects’ knowledge through the life-cycle of the development of the software project. The life-cycle of software development projects can be defined using the systems development life-cycle approach as shown in Figure 2. Figure 2 also shows the various types of knowledge that needs to be managed during the project life-cycle. It has been observed that the following project-related critical knowledge needs to be managed as the project progresses:

- user requirements knowledge
- functional domain knowledge
- technical knowledge

![Figure 2: Knowledge Areas during Systems Development Life-cycle](image)
• project status knowledge
• project experience knowledge.

Need for Managing User Requirements Knowledge

Meeting the client’s requirements is critical to a software project’s success; in fact, it is more critical than meeting time and budget estimates (Bennatan, 1995). Yet, ascertaining those requirements is not easy (Redmill, 1997). Clients may be unable to articulate their requirements. They may articulate the wrong requirements. Besides, different client groups may disagree over requirements. Their articulation of requirements may be misunderstood by the software developers. As a result of this and environmental volatility, requirements may change during a project. This uncertainty may lead to conflict, delays, cost over-runs, and failure to meet the client’s needs.

Requirements refer to the descriptions of properties, attributes, services, functions, and/or behaviours needed in the software to accomplish the goals and purposes of the system (Carr, 2000). At the system level, requirements should address the needs but should not specify a design solution. This should be left to the software designers in the team.

Requirements may be classified as normal requirements which is what the team can get by just asking customers what they want and expected requirements which are considered so basic by the customers that they may not think of mentioning them — until the project team fails to deliver them (Zultner, 1993). Capturing of software requirements seems like a simple matter that can be fast achieved and then put aside. But, as Carr (2000) states, poor requirement specifications are the most significant contributor to software project failures such as software rework, late delivery, over-budget, and either poor or incorrect performance. The main problems with the requirements analysis could be: requirements not reflecting the actual need of system users; inconsistent requirements; incomplete requirements; conflicting requirements; misunderstood requirements; ambiguous or vague requirements; introduced requirements; spurious requirements; and unintended consequences.

One of the reasons for these requirement problems could be that during requirements analysis phase, a lot of information may be presented to the requirements analysts which are never captured by them. Thus, adopting a knowledge management perspective of requirements, Walz, Elam and Curtis (1993) make some specific recommendations for software project managers:

• Increase the amount of application domain knowledge across the entire software development team.
• Actively promote the acquisition, sharing, and integration of knowledge within a software design effort through team facilitation techniques and formally recognize these activities by allocating time to them.
• Much of the information that needs to become part of the team’s memory is not captured formally, particularly, in standard documentation. Therefore, new tools (such as intranets) are needed to easily and unobtrusively capture this process-based information.

Need for Managing Technical and Functional Domain Knowledge

Knowledge from multiple technical and functional domains is a necessity for software development. This knowledge falls along at least three inter-dependent domains (Henninger, Lappala and Raghavendran, 1995):

• application domain such as manufacturing, banking, transportation, etc.
• technical domain
• best practices in the two domains.

Ideally, a software development team should be staffed so that both the levels and the distribution of knowledge within the team match those required for the successful completion of the project. However, in most organizations, this is seldom the case because of knowledge shortfalls such as the thin spread of software application domain knowledge. In general, individual team members do not have all the knowledge required for the project and must acquire additional information before accomplishing development work. The sources of this information can be relevant system documentation, formal training sessions, the results of trial-and-error behaviour, and other team members. Group meetings provide an important environment for learning since they allow the team members to share information and learn about other domains relevant to their work.

Productive software design activities need to resolve around the integration of the various knowledge domains. This integration leads to shared models of the software problem under consideration and potential solutions. A software development team seldom starts
its life with shared models of the system to be built. Instead, these models develop over time as team members learn from one another about the expected behaviour of the application and the computational structures required for producing this behaviour (Walz, Elam and Curtis, 1993). Thus, knowledge acquisition, sharing, and integration are all activities that enable a software development team to learn what it needs for producing an appropriate software design. The length of time that a team spends in its learning phase depends on the breadth and depth of knowledge that the team members bring to the project. It is also affected by the extent to which customers understand the requirements of the project.

**Need for Managing Project Status Knowledge**

The third type of project knowledge, which must be available to the software team, is project status knowledge. Project documentation (such as requirements specification, design documentation, programme specifications, project plans, etc.) and standards (such as checklists, templates, standard procedures, etc.) need to be managed. Documentation embodies know-how; therefore, it should also be stored online so that it could be shared among the entire development team and other peers. Thus, the project-related knowledge that needs to be managed includes project issues, best practices, and project documentation.

The term project issues refers here to the obstacles that arise during the project and threaten to disrupt the progress of the project and issue management refers to project management by responding to the various issues (Glass, 1998). These issues may be related to schedule or project progress, resource requirements, project cost, product quality, development performance, technical adequacy, etc.

**Need for Managing Project Experience Knowledge**

Although issues are always project-specific, they may be having some generic patterns. Therefore, many of the issues encountered in a project could be relevant to other project sites or other projects as well. For example, the issues may be pertaining to important system requirements, instructions or clarifications for customers, innovative design ideas for addressing some problems, precautions to be taken when using some software for development, etc. If such issues can be stored in an issue database, they could be used in the later projects and could help in reducing the project cost, enhancing software quality, or both.

Moreover, software is being developed for an increasingly diverse set of applications and user populations each with different characteristics and development constraints. It is, therefore, becoming increasingly important to manage the application domain knowledge as well so that it can also be applied in subsequent projects. The most valuable kind of knowledge is the one that can be characterized as ‘lessons learned;’ mistakes that developers do not want to repeat, tips and techniques for accomplishing specific tasks, methods that proved to be successful, etc. These lessons form another type of application domain issues and this knowledge should also be created so that it need not be rediscovered (Henninger, 1995). Therefore, such knowledge could be acquired by the development team and stored on the issue database (if there is one) for later use.

**GLOBAL SOFTWARE TEAM KNOWLEDGE MANAGEMENT FRAMEWORK**

This section focuses on process approach to managing the various knowledge areas. A five-layered framework of team knowledge management (Eppler and Sukowski, 2000) has been applied to understand the knowledge management process in global software teams. This framework is applied after carrying out an exploratory study using three case studies.

We studied three project teams of three different IT organizations by carrying out structured face-to-face interviews with the project teams which included senior project managers, team leaders, team members, and quality managers. Two of the companies are global multinational organizations and have off-shore development centres in India. One company operating out of India has global clients. The three case studies bring out the similarities of knowledge management practices in global software teams.

Through unstructured interviews, we identified five types of knowledge as critical knowledge areas which need to be managed — application domain knowledge, technical knowledge, application requirements knowledge, project status knowledge, and project experiential knowledge. Similarly, we identified three types of knowledge bases — project documentation, best practices documentation, and ‘issue bases’ that have been put in place by IT organizations which help in the process of
managing critical knowledge areas. All the 13 respondents accepted that the use of one or more types of knowledge database helped in creating knowledge or contributing towards process improvement, learning, and innovation. The process approach to managing various types of knowledge has been addressed with the help of different layers of the framework.

The Platform Layer

Global software teams work in a virtual space (the platform layer of the model). This virtual team space facilitates knowledge within the team and between teams. It predominantly uses asynchronous communication and IT-based tools and relies heavily on Internet-based communication like e-mails, accessing the common repository through Net, chatting, Net meetings, etc. Apart from this, periodic teleconferencing with the clients and archiving the minutes of the meetings for future reference is also a commonly used form of communication. Archived e-mails are used as evidence for any kind of discussion at a later stage. All the customer interactions through e-mails are kept at a particular location so that any time one can refer to them.

Video-conferencing is not very popular mainly due to technical problems encountered because of which the required quality of picture and voice is not achieved. Net meeting and other collaborative groupware technologies are also found to be used by a limited number of team members. As the client team is situated at a different location in a different country, only a very limited face-to-face interaction takes place. On the whole, interaction with the client is not frequent. It mainly takes place at the time of the project kick-off and later on at the implementation stage. Face-to-face interaction, when it takes place, is limited to the project manager, business development managers, and the client team. The complete development team which resides in India does not get a chance to meet the client team.

As a result, the user requirements knowledge is not captured completely. Archiving the communication and use of ‘project documentation’ as reference or anchors during communication with the team members have helped the team members to communicate within the global software teams. Also, well-defined channels of communication are in place which helped in reducing the ambiguities associated with communication.

To improve the requirements knowledge, a number of team members who were involved in the project was asked to sit as silent observers while the teleconferencing was going on with the client. However, there existed a gap in managing the user requirements knowledge as per the satisfaction of the global team members and it was felt that new tools were required to align the user requirement with both the technical and functional domain requirements related to the project. In some projects, the clients tried to embed their members in the software teams in India to overcome the above problem.

The platform layer helped in managing the project status knowledge to a large extent. Web-based project management software gave the details of the project status and helped in tracking, monitoring, and controlling the progress of the projects.

Domain knowledge and technical expertise knowledge have been addressed through project documentation. It is managed at the time of team building. But, once the project is in progress, best practice project documentation database helps the team members to update themselves with the latest advances in technical and functional areas. Teams are exploiting the communication technologies to enhance their learnings and making effort to capture project experience knowledge through project reviews in project documentation databases.

The Norms Layer

The second layer of the model addresses issues related to team rules, conventions, and general norms. For global software teams, shared team norms are a crucial factor in fostering effective team knowledge management. However, all the companies are either ‘CMM (capability maturity model) level 5’ certified or ‘ISO 9000 certified’ and hence have project documentation in place. Besides, they rely on sharing of knowledge stored in these documents from time to time to solve the problems and/or issues raised by the clients. The quality management system (QMS) of the organization takes control of the documentation process and verifies the same. These documents also help the new entrants to the team in knowing about the history of the project. A senior quality analyst from one of the ISO 9000 companies stated:

The ISO requirements necessitate that we fill the documents for all the activities and then we make them available to all the employees through our central repository. Even if it is a version, it is updated in the central repository and communicated to all. We have a template for a test plan...
and if the customer wants to know the same, we mail it to him.

Project documentation and best practices documentation have helped the team members to manage the project status knowledge to a larger extent and technical and functional domain knowledge to a limited extent. The organizations have been focusing on creating knowledge in the field of cutting-edge technology or new technology through best practices documents and even motivating the employees by rewarding the best contribution, etc. To quote a quality analyst:

If the customer wants us to follow something else, we follow that. If the customer wants some practice, we may combine both and come out with a best practice template.

Another respondent, who was a project manager in a global multinational company, stated:

We have a best practice repository (BPR). It is only after an item is characterized as best practice (following a discussion) that it finds its place in BPR which is then intimated to the team.

Also, the use of ‘issue databases’ (using software such as Lotus Notes) by software teams for capturing the issues encountered in the project was found to be prevalent. Five of the 13 respondents mentioned using the issue database as reference in communication. As a senior project manager stated:

Before any meeting, there is an agenda of the issues to be resolved and published to the members who would participate in the meeting including the customer.

Another information analyst from a global multinational company stated:

There are different forums and there are different meetings. We have status meetings and there are fixed agenda. We go by the agenda and review the status of CR (change requests), open issues, close issues, log all the information, and refer to them as and when required.

However, the issue of managing experiential knowledge is still at large and open. There are no indications that the knowledge captured in these issue databases could be used in other projects with the same client or another client with a similar project. As pointed out by a senior project manager:

Generally, it is knowledge generated by a small group, say ten persons, which gets shared by 180 people. We, at the mobility and network department, are definitely sharing it but whether others are also sharing it, only quality department could tell.

Peers from around the world could access the issue knowledge base and other knowledge base from different angles like the nature of the problem, modules, people or location. As a Chief Technology Officer stated:

The knowledge base (KB) not only serves as a body of knowledge that we can refer to but also improves the learning. For example, if I am outside the country in a particular area, I may not have the knowledge pertaining to a particular project but I would have to immediately access the information and talk to the customer at the same level; at the same time, I may not want to disturb the project members due to the time zone difference. The customer may ask: “Have you done this before?” As a project manager, I do not know the details of each and every project that my organization has done in the past five to ten years. However, I can refer the KB, search on keywords, and take a print-out and give it to the customer. Regarding the learning part, it is captured very frequently in the KB. For example, after every project is completed, KB is updated. If the technical team comes across good information, KB is updated. Each one of us has expertise in a particular area and we are always keeping abreast of the latest developments in that area; for example, white papers are added to the KB very frequently. So, the data that goes into the KB are unstructured but the KB management system provides a structure to it.

It was found that efforts are made in the direction of managing the experiential knowledge and capturing the same in the database. As another senior project manager stated:

The USP of having a complete documentation process is that every project starts on the strengths of the previous project. The new project builds on those learnings while improving upon them. For example, earlier, we were not documenting the minutes of the meetings or discussion documents of telephonic conversations. These were introduced later. For every new project, we have QMS guidelines tailored for that particular project, and, as soon as the project is initiated,
we start following the guidelines. End-of-the-project learning report and documents are submitted to the quality system. They are reviewed by the quality team and are formalized as new guidelines for future projects.

We found that all the 13 respondents accepted that they have used one or more types of knowledge bases which helped in creating knowledge or contributing towards process improvement, learning, and innovation but only to a limited extent.

**The Processes Layer**

The exploratory study indicated that a few of the knowledge management processes have been put in place but these processes are at a very low level of maturity. Though the CMM process improvement methodology does not include any knowledge management key process activity (KPA), still, CMM implementation in organizations has helped the team members to address some critical knowledge areas. Critics of CMM methodology also believe that CMM structures may encourage a rigid bureaucracy that can stifle creativity and innovation. Thus, by not focusing on knowledge sharing, ‘fertile fields’ for cultivating innovation are not possible. Realizing this, many organizations have started encouraging knowledge management by introducing knowledge management processes in the organizations. As per the layered knowledge management framework, the goal of knowledge auditing process is to make the present team knowledge transparent and discover knowledge deficits in the team for the tasks at hand. Though the software team is assembled as per the availability of manpower and also the interest of team members, in case they lack the technical know-how, the capacity of the team knowledge auditing process would be limited for providing appropriate training to them. There is no indication whether a team member’s professional background and experiences are shared among the other team members.

The CMM structure requires that the team captures the ‘knowledge artefacts’ generated during the progress of the project in various knowledge databases like project database, issue database, and best practices database. As these databases are stored in a shared space and the team members have access to the same, most of the time, they have to brief themselves about the knowledge stored. At the initiative of a few project managers, sharing does take place through presentation sessions but this is largely due to the effort of individual project manager and institutionalization of such processes is yet to take place.

The quality management system of the organizations is making efforts to not only store these knowledge artefacts but also keep them updated and promote their use in the projects by the team members. Thus, there are efforts towards capturing the individual insights as best practice and sharing it across the organization. It is noticed that efforts are made to motivate the team members to replicate successful and best practices. The databases and rules facilitate project status knowledge and domain and technical knowledge to a large extent. These norms also help in narrowing the gap of understanding the user requirements knowledge. Project experience knowledge is also captured in the form of project review databases.

Eppler and Sukowski (2000) discuss about four crucial knowledge management processes: team knowledge auditing, team knowledge development, mutual updating, and briefing and reviewing. CMM implementation has helped in implementing software process in the organizations; however, the knowledge management processes are introduced on an *ad hoc* basis.

Thus, team knowledge auditing, development, updating, and briefing and reviewing processes have found a place in the working of global software teams but these processes are *ad hoc* and immature. They are limited to individual projects and have not yet evolved across the organization. At the team level, though a lot of knowledge sharing can take place between the team members, whether members of the teams of different projects gain from this is still questionable. Our study showed that sharing across the projects is not yet taking place at the organizational level.

**The Tools Layer**

The tools for team knowledge management focus on the elicitation, categorization, and aggregation of dispersed knowledge in a team. Their main goal is to make knowledge in its various forms more transparent for every team member. Our study did not focus on identifying the existing tools of knowledge management.

**The Leadership Layer**

Knowledge management also requires a leadership commitment with the leader serving as a role model. In this study, leadership commitment has not been studied. Hence, this study falls short with respect to these two layers of the knowledge management framework.
CONCLUSIONS AND RECOMMENDATIONS

The business drivers of any IT organization are competitive responsiveness, collaborative work culture, shorter time-to-market, and tacit knowledge. CMM certification and Six Sigma certification are efforts in the direction of improving the software processes. However, while these certifications have helped in improving the software processes, the critics believe that CMM structures encourage rigid bureaucracy that can stifle creativity and innovation. CMM certification does not include key process activities directly addressing the issue of knowledge management. However, the entire concept has helped software teams improve the project status-related knowledge and creation of databases.

Our study identified five critical knowledge areas of global software teams that should be managed — user requirements knowledge, functional domain knowledge, technical knowledge, project status knowledge, and experiential knowledge. Out of these, the study reveals that project status knowledge was well managed. There were well-defined processes in place which were also adhered to by the team members. These processes had also matured over time, one of the reasons being that two of the organizations were CMM level 5 certified and one was ISO 9000 certified which depicts the higher level of maturity in following the software development processes.

Technical knowledge related to the project was indirectly managed by providing training to the team members when they were assigned to a project and a gap was identified in terms of technical knowledge required for that project. Also, best practice documentation helped the team members to learn and update their knowledge base. This was the practice followed by all the teams which can be attributed to the high turnover of team members and customers wanting to work with the latest technology which is changing very fast. It was found that, of late, more domain knowledge specialists have been employed and embedded in the software teams at the organizational level. These specialists are also required to manage the user requirements knowledge as they are responsible for converting the user requirements to design requirements. But, there exists a gap and there are instances of user requirements not being met by the project team as a result of which they had to re-work. Still, no tools or scale have been developed to measure this and hence we can at best say that this knowledge is also managed informally.

The team members have shown excitement towards the need to manage the experiential knowledge but have not addressed the same in a systematic manner. Thus, the issue of managing the critical project knowledge has to be addressed in a systematic way and processes and tools to implement the same have to be formalized and institutionalized.

As per the knowledge management framework, all the identified knowledge related to the project are not well managed due to the absence of knowledge management processes in the teams. Though the platform and the norms layer of the framework are well supported by software teams, which supported the knowledge areas, the knowledge processes were ad hoc. This study falls short of commenting on the tools and the leadership layer of the framework.

REFERENCES

Duffy, J (2000). “Knowledge Management: To Be or Not To Be?” Information Management Journal, 34(1), 64-68.
Sangeeta Shah Bharadwaj is Associate Professor of Information Management at Management Development Institute, Gurgaon. She is a Ph. D. from Birla Institute of Technology and Science, Pilani. She has taught at BITS, Pilani, worked as a software engineer in the US, and successfully headed and managed an entrepreneurial venture 1-800-FLOWERS franchise there for four years. She has several publications in *OPSEARCH, Indian Management, and JITCA*. Her teaching and research interests include MIS, managing IT projects, software project management, information technology and business process outsourcing, and knowledge management.

e-mail: ssbharadwaj@mdi.ac.in

Kul Bhushan C Saxena is Professor of Information Management at Management Development Institute, Gurgaon. He is a Ph.D. in Management from Gujarat University, Ahmedabad. Prior to joining MDI, he taught at the Erasmus University Business School, Rotterdam, the Netherlands and the Hong Kong Polytechnic University for 13 years. He is a member of the editorial board of *Journal of Information Resources Management*. He has won, among others, the ‘Hong Kong Computer Journal Best Paper Award’ of the Hong Kong Computer Society and the ‘Distinguished Achievement Award’ of the Indian Space Research Organization. His research, teaching, and consulting interests are in e-governance and public administration reengineering, e-business strategies and process management, and enterprise systems for small businesses. He is a member of the British Computer Society and Association of Computing Machinery, USA. He is the Chairman of the Centre of Excellence in Information Management (which has been set up with funding from the European Union) as well as that of the Information Management Area. He has published more than 100 papers and is the co-author of a book titled *World-class Manufacturing* with Dr B S Sahay.

e-mail: bsaxena@mdi.ac.in