2010 International Conference on Global Software Engineering

Crafting a Global Teaming Model for Architectural Knowledge

Sarah.Beecham, John Noll, Ita Richardson, and Nour Ali
Lero—The Irish Software Engineering Research Centre
University of Limerick
Limerick, Ireland
{Sarah.Beecham, John.Noll, Ita.Richardson, Nour.Ali}@lero.ie

Abstract—**In this paper, we present the Global Teaming Model (GTM), which is empirically grounded, and outlines practices that managers need to consider when managing virtual teams. We explain how the model can be adapted to specific areas of software development, and use architectural knowledge management (AKM) as our exemplar. We focus on specific practices relating to how teams collaborate and share essential architectural knowledge across multiple sites. Through a review of the literature, we develop an in-depth view of recommended practices associated with AKM in a global environment. We then consider how we can incorporate these AKM practices into our Global Teaming model to ensure managers are given the necessary support. Our contribution to research therefore is to present AKM practices within the context of all other Global Software Development processes.**

Keywords-Global software development, Distributed Software Development, Global software engineering, Architectural Knowledge Management, Software Processes; Virtual Teams; Software Process Improvement; Global Teaming Model

I. INTRODUCTION

It is becoming more commonplace for software engineers to work in a globally distributed environment [20] – which we call Global Software Development (GSD)[1]. GSD has complexities over and above those experienced in local software development [7, 10, 17, 22] and we can discuss 'global distance' from four viewpoints: geographical, temporal, cultural and linguistic, with each causing specific issues for software engineers working in this environment. Geographical distance introduces physical separation between team members and management [6]. Temporal distance hinders and limits opportunities for direct contact and cooperation [1]. Cultural distance negatively impacts on the level of understanding and appreciating activities of colleagues and teams [9, 31]. Linguistic distance, usually identified through the lack of a common native language, causes communication problems [8, 20, 25].

[1] A variety of terms exist: Distributed Software Development (DSD), Global Software Engineering (GSE), and Global Software Development (GSD). We will use the term GSD in this paper.

In global software development (GSD), increased dependence on the architecture goes hand in hand with increased complexities inherent in a virtual working environment, where time zones, culture and languages may differ, and geography may prevent teams from meeting face to face. It is important to have processes in place to manage the architectural knowledge, especially since it is likely that there will be increased communication difficulties when operating in a global environment.

To achieve improvements in an organization's architectural capabilities, the architecture knowledge needs to be managed [2]. In this paper, we consider how this architectural knowledge can be captured and managed through an implementation of an established 'Global Teaming' model (GTM) developed for virtual teams [28, 29]. We do this in two steps; first we take a systematic approach to selecting Architectural Knowledge Management (AKM) practices (with a view to integrating them into our GTM); and secondly by examining how the defined processes in our GTM facilitate AKM. The Global Teaming Model is derived from empirical work and aims to cover all the specific needs of the virtual global team [28, 29]. We now demonstrate how the model can be implemented in order to support those practices associated with AKM in a global environment.

We take a process view of how architectural knowledge is managed, since "Architectures, plans, and processes are all vital coordination mechanisms in software projects" [21]. Global distance can interfere in a variety of ways with the effectiveness of a project team's communication [8, 20, 21, 25].

To summarise, in this paper, we discuss the importance of managing AKM in a distributed environment through an identification of key practices sourced from the literature, and then consider how these AKM practices can be implemented through an augmentation of the GTM. The purpose of the paper is to validate the GTM, and we use AKM specific practices to do this, as AKM is recognized as an important process in Global Software Development. Our research question therefore is, "Can the Global Teaming Model be used to underpin specific practices associated with AKM?"

The paper is organized as follows: We first introduce the Global Teaming Model in Section II; in Section III, we

978-0-7695-4122-8/10 $26.00 © 2010 IEEE
DOI 10.1109/ICGSE.2010.15

give a brief background to Architectural knowledge management. Section IV explains our method relating to the literature review. In Section V, we present our results by listing practices identified in the review of the literature. We then combine the Global Teaming Model with the AKM practices in Section VI and discuss some limitations to the study. We conclude our study with a summary of our contributions in Section 0.

II. Overview of the Global Teaming Model

Lero—the Irish Software Engineering Research Centre, has been carrying out research into GSD for a number of years. During various studies, we have observed that the management of global teams in process models such as the CMMI® is not explicitly defined. Given the substantial growth of GSD, we addressed this gap through the development of an additional process area called Global Teaming (GT) [29]. GT establishes goals and sub-practices specific to GSD, that are not addressed in the CMMI® model. While we have structured this process area to be similar to those in the CMMI® model, the Global Teaming process area does not require CMMI® implementation for its use. The process area is relevant to the continuous model for CMMI® as opposed to the staged representation as the continuous model allows the organisation to select those process areas that provide most benefit to them. The Global Teaming process area can therefore be used with other CMMI® process areas, or as a stand-alone process model which organisations can implement when establishing global software teams; hence, we refer to it as the Global Teaming *Model* (GTM) to emphasize its stand-alone capability.

Lero researchers, including two of this paper's authors, developed practices for the Global Teaming Model by identifying issues that can affect the management of global teams as presented in the literature and our empirical research. In defining the Global Teaming Model, we are only interested in sub-practices which relate specifically to the global situation. For example, the CMMI® views Project Management in terms of sub-processes such as Project Planning; Project Monitoring and Control; Supplier Agreement Management; Integrated Project Management; Risk Management; Integrated Teaming; Integrated Supplier Management and Quantitative Project Management [15]. As such, when defining the Specific Practice SP1.3 "Global Project Management", we only note sub-practices over and above the project management sub-practices aimed at collocated teams already present in the CMMI®.

The GTM contains two Specific Goals – Define Global Project Management (Figure 1) and Define Management between Locations (Figure 2). Each practice in our Global Teaming Model was developed directly through an analysis of our primary and secondary research findings [28, 29].

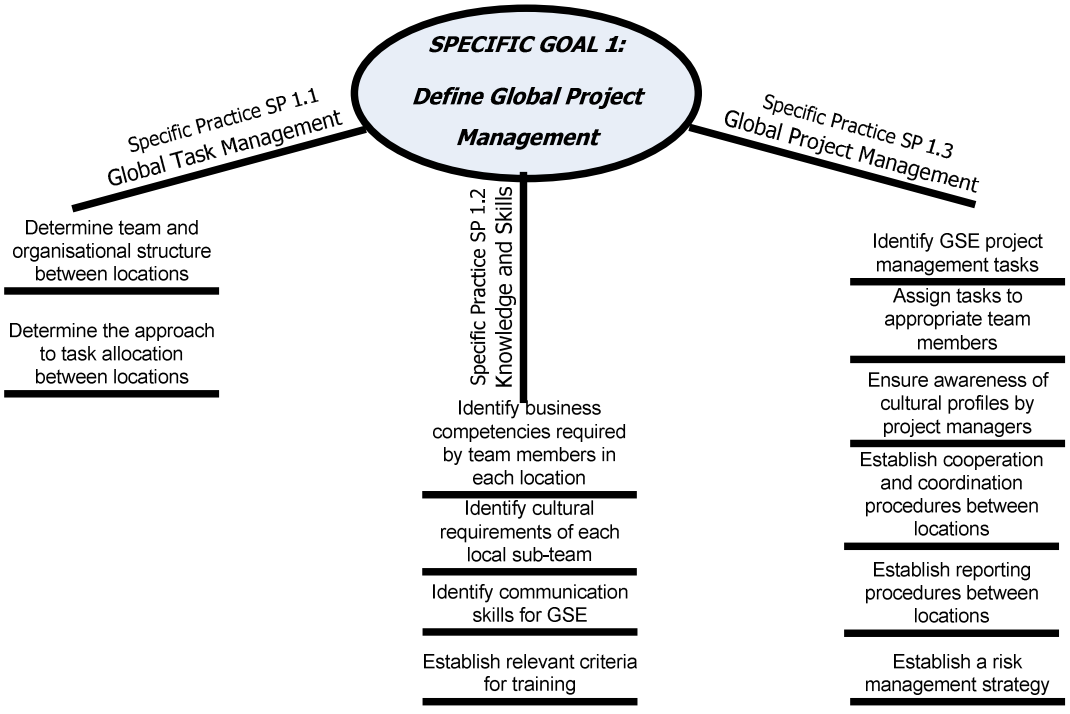

Figure 1: Specific Practices relating to Goal 1 in the Global Teaming Process.

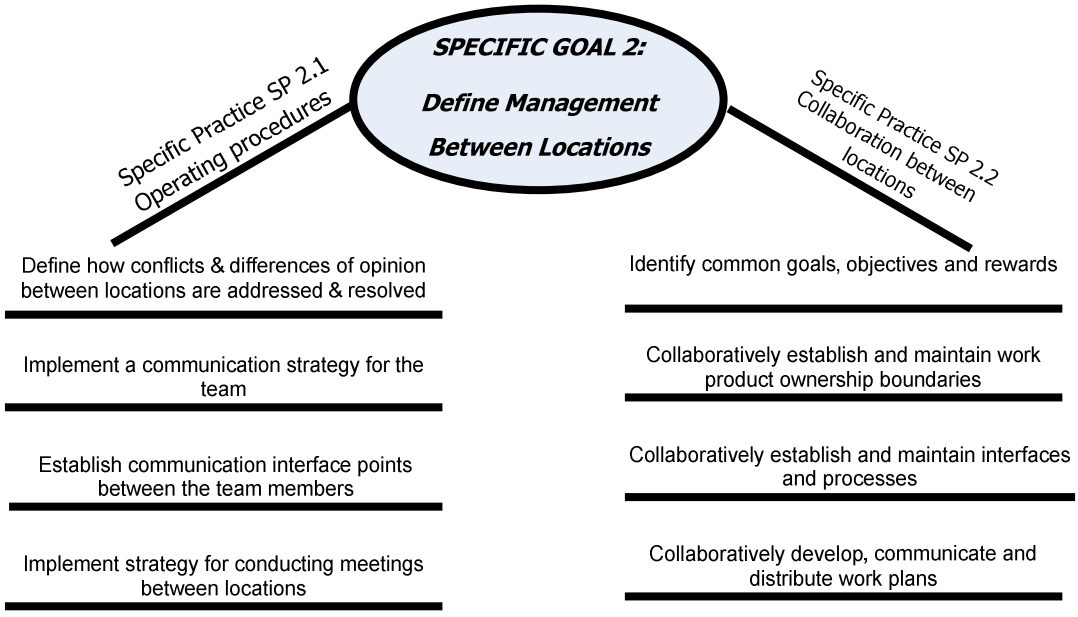

Figure 2: Specific Practices relating to Goal 2 in the Global Teaming Process.

The research in this current paper is interested in defining practices within the GTM Specific Practice SP 2.2 "Collaboration Between Locations" (Figure 2) in connection with AKM, which ensures that team members can collaborate across global boundaries. Therefore, in this section we discuss the sub-practices in more detail and demonstrate how the practices in the GTM underpin the more detailed and prescriptive practices associated with AKM.

A. Collaboration between Locations

The Global Teaming Model defines Specific Practice 2.2 "Collaboration Between Locations" comprising four sub-practices, as follows:

(1) Identify common goals, objectives and rewards for the global team.

It is important when setting up global teams that particular factors are taken into account. While global teams require goals and objectives to be agreed and understood by all the team members, regardless of location, the achievement of these goals should be measured jointly across locations by their accomplishment [23].

(2) Collaboratively establish and maintain the work product ownership boundaries among interfacing locations within the project or organisation.

Although the global team should be regarded as a single team, when defining the work to be carried out in each location, it is important to establish the work product ownership boundaries. This includes explicitly identifying who is responsible for specific tasks within a single work product. It is important to establish a method for the partitioning and the allocation of work across the global team. For example, when requirements changes are identified, they should be distributed to those responsible for interfaces so that interfacing requirements are also identified and modified.

(3) Collaboratively establish and maintain interfaces and processes among interfacing locations for the exchange of inputs, outputs, or work products

One of the requirements of implementing used and useful processes is to ensure that those using the processes have ownership of the processes. Team members can be alienated where processes have been set up or modified without the involvement of those at all sites [13], and this becomes particularly important in the global situation. Oftentimes, project managers at (normally) the parent site, implement this process in distributed sites. This does not give ownership to everyone on the team, and can cause problems when everyone does not follow the implemented process. Therefore, the global team should be involved in the development of the interfaces and the processes required for efficient software development.

(4) Collaboratively develop, communicate and distribute among interfacing teams the commitment lists and work plans that are related to the work product or team interfaces.

In GSE, not only do distributed software teams need to agree achievable milestones, there is a requirement that ongoing progress with reference to costs, time, productivity, quality and risk are overseen. Contingency plans to monitor risks should be implemented. These should include procedures for implementation if they are ever required.

Due to the importance of synchronous and asynchronous communication tools for GSE communication, communication plans should be explicitly included.

Richardson et al. state that global organizations face a real challenge in managing their knowledge [30]. Managing Architectural Knowledge is no exception and we note the importance of explicitly defined processes to support the required communication [5]. While the GTM presents specific goals, specific practices and sub-practices for organizations implementing GSD, a more in-depth definition of the practices is required for process implementation.

III. Architectural Knowledge Management and GSD

Software architecture is a discipline that focuses on the design and specification of overall system structure. Not only does the architecture guide the structure of a system to be developed, but also the structure of a project and of an organization [4]. Architectural Knowledge Management involves managing, capturing and sharing the information produced and consumed during the software architecture process. This knowledge involves the skills and expertise of teams, design decisions, the business drivers, the functional and non-functional requirements. This knowledge is specific to the domain, project and organization.

Architectural knowledge spans knowledge of the problem domain (e.g., architectural requirements, drivers, constraints), the solution domain (e.g., architectural tactics, patterns, styles), and knowledge entities used in architecting itself, including:

- The architectural design;
- Assumptions made during the architectural design and underpinning design decisions;
- Linkage to the environment; design decisions; interdependencies between the design decisions;
- Mapping of design decisions to functional and non-functional requirements, needs, constraints, design and implementation; the domain analysis;
- Architectural patterns used;
- Design alternatives evaluated;
- Rationale [11].

Thus, Architectural Knowledge Management (AKM) supports creation, storage, and dissemination of architectural specifications, decisions, and rationale. This support becomes even more essential in a distributed context where architecture decisions among different sites need to be taken and architectural knowledge needs to be shared among distributed teams. Architectural rules can help to overcome some of the challenges of GSD where "Architectural rules are principles and statements about the software architecture that must be complied with throughout the organization" [12].

In Global Software Development, the role that software architecture plays in bridging the gap between requirements and implementation becomes even more important for achieving quality systems [19]. Previous research has indicated that there is inter-dependence between the architectural structure and the organizational structure [16, 32] where communication takes on an essential role. It is therefore clear that when software organizations off-shore or outsource parts of their development, the architecture can help to define the organizational structure and consequently the channels of communication. Also, "an architecture-approach can bring about systematic management and productivity for conventional processes"[19]. Faria suggests that a common architecture that orientates all software life-cycle processes, can favorably influence GSD business [19]. Such recommendations can be widely found in the literature. This provides a rationale for our study, as it is essential to support the practices needed to communicate and exchange architecture knowledge, coordinate the groups, the activities and artifacts involving the distributed architecture process, and manage architectural dependencies among tasks.

IV. Method

This section explains briefly how, through a systematic investigation of the literature, we identified key practices associated with AKM in a global environment. Through a review of the literature, we seek to answer the following research question:

What are the recommended AKM practices for Global Software Development?

This question is formed specifically for the literature review and underpins our key research question which is to use the identified AKM practices to validate the Global Teaming model.

For this study, we have taken a focused yet systematic approach to identifying research publications relevant to our research question. We do not aim to uncover *all* the recorded practices, but to select a sufficient collection of studies that allow us to identify recurring themes in a cross section of studies. This methodology is very similar to that used in [27].

The following steps are recommended from systematic review guidelines [24]:

1) Identify the need for a systematic literature review.
2) Formulate review research question(s).
3) Carry out a search for relevant studies.
4) Assess and record the quality of included studies.
5) Classify data needed to answer the research question(s).
6) Extract data from each included study.
7) Summarise and synthesise study results (meta-analysis).
8) Interpret results to determine their applicability.
9) Write-up study as a report.

This study conformed largely to these guidelines, with some modifications as discussed below.

Need for a review.

We have previously undertaken an extensive empirical study that focused on problems encountered in GSD [29]. One of the outputs of this in-depth study was a Global Teaming model (introduced in Section II), where we define what is required to collaborate in a virtual environment. Subsequently, we have examined the software engineering literature and have not found a comprehensive survey that addresses our research question; hence, the need for this review.

Search.

We used the following Boolean search string to ensure we captured a wide variety of papers that related to practices in AKM in global software development:

"Global Software" AND "Architect*"

We used this string to search the *IEEEXplore (http://ieeexplore.ieee.org)* bibliographic database, where using such general terms ensured that we captured studies using terms such as "Global Software Development" and "Global Software Engineering", and "architecture" and "architectural knowledge". The search was performed in January, 2010 and resulted in 30 papers. We then conducted a secondary search based on papers that are cited in our thirty references and key GSD and AK conferences and workshops to include:

- SHARK http://www.cs.rug.nl/~paris/SHARK2010;
- ICGSE http://www.icgse.org/;
- KNOWING http://www.lero.ie/knowingworkshop/

Document selection.

Inclusion and exclusion criteria were used to select the subset of papers from those identified by the initial search, that should be included in the analysis of the research question:

We include texts that:

- directly answer our research question
- present lists of practices that capture everything required to manage AKM in GSD (not just one or two practices, or a validation of a specific tool).
- represent empirical observations
- are full research papers, peer reviewed, published in a journal or conference proceedings (e.g. not an editorial or introduction to a workshop, or book chapter).

Before accepting a paper into the final set for review, we checked for repeated studies to ensure there is no duplication; for example if the same study is published in two different journals with different first authors, only one study would be included in the review; usually the most comprehensive study or the most recent study.

This process resulted in a selection of six studies [3, 11, 18, 20, 26, 32] that we used to identify practices in AKM.

Kitchenham recommends using a study quality assessment checklist to assess the quality of studies for inclusion in a systematic review, including the quality of the research method [24]. Since the current study is an attempt to identify themes, rather than establish statistically valid conclusions, the quality criteria for inclusion in the current study are straightforward, so we did not create such a checklist.

Data extraction, meta-analysis, and interpretation.

We examined each selected study to extract identified lists of practices relating to AKM in GSD (see Section V);

Then, we synthesized the data by first identifying major categories of AKM practices in each selected paper.

Subsequently, a summary was created showing the theme and the paper(s) that identified the theme (see Table 1).

We give each occurrence the same weight, so the frequencies merely reflect how many times a given practice is identified in different papers, not how important it might be.

V. Results: AKM Practices for Global Teams

Architectural knowledge management practices employed by global teams need to address the challenges of GSD [13]. Our review of the literature demonstrated that challenges for achieving architectural knowledge management fall into three areas:

1. Alignment of architecture and organization ("Conway's Law");
2. Knowledge management practices for creating and disseminating architectural knowledge;
3. Infrastructure for managing architectural knowledge.

As such, in the following sections, architectural knowledge management practices for global teams are grouped according to how they address the three broad challenge areas listed above.

A. Conway's Law

In 1968, Melvin Conway observed that "... organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations" [16]. This observation has implications for architectural knowledge management practices.

Herbsleb [20] has identified three challenges that have to be tackled for architectural knowledge management in global software development; these challenges follow directly from Conway's law:

1. To understand the knowledge about the relation-ship between software architecture dependencies and task dependencies;
2. To assess how an organization is prepared to carry out the design and implementation of an architecture;
3. To provide tactics that adjust the organization to the architecture and vice versa.

Laredo and Ranjan [26] suggest that the architecture should be decomposed into components according to

geographic distribution of teams, to provide "vertical" allocation of functionality. Salger [32] echoes this view; in addition to an architecture that matches the organizational structure, Salger advocates fine-grained modularity so a given module can be developed at a single site.

Avritzer et al. [3] also suggest that "components are allocated to individual teams, and the components interact with each other through well-defined interfaces". They note that a system's architecture should be consistent with the organization's coordination structure, to facilitate collaboration. To achieve this, they recommend an organizational approach where architects have *cross-team* communication responsibilities, while developers have *intra-team* responsibilities. This reduces overhead and delay by concentrating communication with local peers rather than remote personnel. This also means that each site should have an architecture lead, so that developers have a local resource to consult regarding architectural issues.

B. Knowledge Management Practices

Alignment between the architecture and the organizational structure is a necessary, but not sufficient, condition for effective dissemination of architectural knowledge. Based on a review of the literature related to architectural knowledge management, Clerc identifies *practices* considered essential to effective architectural knowledge management [11]:

1) Frequent interaction across sites, preferably via on-site visits.
2) Cross-site ``delegation'', where a site allocates personnel to relocate temporarily to remote sites, to establish shared understanding of the system architecture, and to create personal relationships between the local and remove sites.
3) Face-to-face kickoff meetings, to establish personal relationships across sites and ensure all sites have the same expectations.
4) Provide a "network of volunteers" who can answer questions about architectural alternatives, design designs, etc. However, there must be sufficient information about the roles and expertise of the members of the network in order for the architects to trust their responses.
5) Develop the high-level architecture via an initial co-located design session, to achieve a "sound high-level architecture".
6) Implement clear organizational structure to provide clear lines of communication among stakeholders at local and remote sites. This requires clear identification of stakeholder roles and their responsibilities, as well as technology to support communication.
7) Create a repository for artifacts, to store decisions, rationale, and the actual architectural designs.

C. Knowledge Management Infrastructure

In a global software development environment where geographic and temporal distance impede face-to-face interaction, adequate communication infrastructure must be deployed to support architectural knowledge management practices, as well as to capture design decisions, artifacts, and rationale. Of particular importance is a record of how architectural decisions are consistent with requirements [26, 32]. Toward this end, a Wiki can be used to capture documentation of how the architecture meets requirements, as well as both facilitate and record discussion of design alternatives [26].

In addition, Farenhorst et al. [18] determined that architects need a single portal for accessing architectural knowledge that is related to both a given project, and to the organization as a whole. This portal should provide search capabilities, and event notifications to keep stakeholders up-to-date on changes to particular artifacts as well as the organization as a whole [18, 26].

VI. Implementing AKM Practices

This section combines the AKM practices identified in Section V together with the defined processes in the Global Teaming Model (GTM) (see Figure 2). Whereas all practices listed in the GTM are likely to be of relevance to the management of GSD, for AKM we focus on the practices relating to Specific Practice 2.2 "Collaboration between Locations". Table 1 summarises our results. It shows that all practices listed in the six AKM studies relate to the more general practises listed in the GTM. Table 1 also indicates that each of the six studies has something new to say about how to manage AKM in GSD when examined at this more detailed level. Yet there are many overlapping themes, and the most relevant practice appears to be "*Collaboratively establish and maintain work product ownership boundaries"* where we identified several sub-themes such as *AKM and Organisational Structure*; *Task allocation*, and *Roles and Responsibilities*

A. Limitations of the Study

Internal validity: We cannot be certain that we have covered *all* AKM practices, only those that have been listed in the key papers have been included. However, the practices we identify allow us to meet our aim which is to provide an example of how the GTM can be used to frame and think about practices at a level that can be implemented. Our representative sample gives us confidence in the practices recommended. For example, those recommended by Clerc [11] stem from a review of the literature and an empirical study that is further validated in [13]. We use this as a main input to our model. All the practices we list are extracted from empirical studies.

.

Table 1: AKM practices in context with the Global Teaming Model

Collaboration sub-practice 1: Collaboratively establish and maintain interfaces and processes [For AKM]
Understand the relationship between software architecture dependencies and task dependencies [20]
Architecture-Implementation interface: Assess how organization will carry out design and implementation of the architecture [20]
Requirements-Architecture interface: Ensure significant architecture decisions satisfy 'architecturally significant' requirements [32]
Collaboration sub-practice 2: Identify common goals, objectives and rewards [for AKM]
Hold face-to-face kickoff meetings to establish personal relationships across sites and ensure all sites have same expectations [11]
Collaboration sub-practice 3. Collaboratively establish and maintain work product ownership boundaries [for AKM]
AKM and Organisational Structure
Implement clear organizational structure for clear lines of communication among stakeholders at local and remote sites [11].
Ensure system's architecture is consistent with the organization's coordination structure, to facilitate collaboration [3]
Decompose architecture into components according to geographic distribution of teams, to provide "vertical" allocation of functionality [26, 32]
Task allocation
Allocate components to individual teams [3]
Consider fine-grained modularity so a given module can be developed at a single site [32]
Understand the relationship between software architecture dependencies and task dependencies [20]
Roles and responsibilities:
Cross-site ``delegation'', where a site allocates personnel to relocate temporarily to remote sites, to establish shared understanding of the system architecture, and to create personal relationships between the local and remove sites [11]
Provide a ``network of volunteers'' who can answer questions about architectural alternatives, design designs, etc. However, there must be sufficient information about the roles and expertise of the members of the network in order for the architects to trust their responses [11]
Identify stakeholder roles and their responsibilities [11]
Take an organizational approach where architects are responsible for *cross-team* communication [11]
Take an organizational approach where developers are responsible for *intra-team* communication [11]
Each site should have an architecture lead [11]
Collaboration sub-practice 4: Collaboratively develop, communicate and distribute work plans [for AKM]
Develop the high-level architecture via an initial co-located design session, to achieve a ``sound high-level architecture'' [11].
Identify technology to support communication [11].
Encourage frequent interaction across sites, preferably via on-site visits [11].
Create a repository for artifacts, to store decisions, rationale, and the actual architectural designs [11]. Architects need a single portal for access to architectural knowledge related to both a given project, and to the organization as a whole [18].
Ensure that the architectural lead is available for consultation with local developers [11].
Record discussion of design alternatives and documentation on how architecture meets requirements. Wiki recommended to facilitate this form of communication [26].
Create a central repository with search capabilities, and event notification to keep stakeholders up-to-date on changes to particular artifacts as well as the organization as a whole [18, 26].

External validity: We cannot tell if all the listed AKM practices will be effective for all global teams. As noted by Herbsleb [20], many authors propose plausible rules of thumb but we need future research to validate whether these views apply to all situations. We have tried to keep the practices generic, but it is possible that they don't all apply; for example, some organisations may prefer to use a purpose-built system as opposed to a WIKI [18]

VII. Concluding Remarks

In this paper, we presented a summary of the Global Teaming Model (GTM) – a model that represents the key practices that software organizations should consider when operating in a geographically distributed environment. The GTM is derived from both direct empirical evidence and an extensive review of the literature. The GTM is a descriptive process model, so to implement the practices would require organizations to tailor them to their own specific needs.

In order to show how this can be achieved, we have taken a specific area of project management, Architectural Knowledge Management (AKM) and have applied the practices to the GTM. Many organizations neglect their systems software architecture due to the fact that AKM is difficult and costly to maintain. Since software architecture is not only used to bridge requirements and implementation but also can be a coordination tool in a global context, we believe that it is important to be applied using our GTM. We believe that mapping AKM practices to our GTM will motivate practitioners to use our model and will provide them with specific recommended AKM practices.

This study therefore contributes to the more general research in GSD, as practitioners and researchers can now view the key AKM practices in context with all other processes required when working with teams across multiple sites. We take a critical look at the practices associated with AKM in a Global Software Development context. We have therefore shown that the Global Teaming Model has the flexibility to be used as a framework for more specific and prescriptive practices. Future work includes applying the model to solve problems associated with the Requirements Process in GSD, and how to manage the Testing function in a global environment. We are currently validating the Global Teaming Model in industry and plan to publish the results in due course.

Acknowledgment

This research is supported by Science Foundation Ireland through Grant No. 03/CE2/I303.1 within Lero - the Irish Software Engineering Research Centre (http://www.lero.ie).

References

[1] P. J. Ågerfalk and B. Fitzgerald, "Flexible and distributed software processes: old petunias in new bowls?" Comm. of the ACM, vol. 49, no. 10, 2006, pp. 26-34.

[2] M. Ali-Babar, D. Winkler, and S. Biffi, "Evaluating the usefulness and ease of use of a groupware tool for the software architecture evaluation process," 1st Intl. Symp. on Empirical Software Engineering and Measurement (ESEM 2007), 2007, pp. 430-439.

[3] A. Avritzer, D. Paulish, and C. Yuanfang, "Coordination implications of software architecture in a global software development project," 7th Working IEEE/IFIP Conference on Software Architecture (WICSA 2008), 2008, pp. 107-116.

[4] L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 2nd Ed., Addison Wesley, 2003.

[5] S. Beecham, T. Hall, and A. Rainer, "Defining a requirements process improvement model," Software Quality Journal, vol. 13, no. 3, 2005, pp. 247-279.

[6] E. Carmel, "Global software teams: collaboration across borders and time zones," Saddle River, NJ: Prentice Hall, 1999.

[7] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," IEEE Software, vol. 18, no. 2, 2001, pp. 22-29.

[8] E. Carmel and P. Tjia, "Offshoring information technology: sourcing and outsourcing to a global workforce," Cambridge, U.K.: Cambridge University Press, 2005.

[9] V. Casey, "Leveraging or exploiting cultural difference?" 4th IEEE Intl. Conf. on Global Software Engineering (ICGSE '09), 2009.

[10] V. Casey, Software Testing And Global Industry: Future Paradigms, I. Richardson and M. O'hAodha, Eds., Newcastle, UK: Cambridge Scholars Publishing, 2009.

[11] V. Clerc, "Towards architectural knowledge management practices for global software development," Proc. 3rd ACM Intl. Workshop on Sharing and Reusing Architectural Knowledge, 2008, pp. 23-28.

[12] V. Clerc, P. Lago, and H. van Vliet, "Global software development: are architectural rules the answer?" 2nd IEEE Intl. Conf. on Global Software Engineering (ICGSE '07), 2007, pp. 225-234.

[13] V. Clerc, P. Lago, and H. van Vliet, "The usefulness of architectural knowledge management practices in GSD," 4th IEEE Intl. Conf. on Global Software Engineering (ICGSE '09), 2009, pp. 73-82.

[14] V. Clerc, P. Lago, and H. van Vliet, "Assessing a multi-site development organization for architectural compliance," Proc. 6th Working IEEE/IFIP Conference on Software Architecture, IEEE Computer Society, 2007.

[15] CMMI Product Team. "Capability Maturity Model® Integration for Development," Technical Report, Software Engineering Institute, 2006.

[16] M. Conway, "How do committees invent?" Datamation, vol. 14, no. 4, 1968, pp. 28-31.

[17] D. E. Damian and D. Zowghi, "An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations," Proc. 36th Intl. Conf. on Systems Sciences (HICSS '03), Hawaii, 2003, pp. 1-10.

[18] R. Farenhorst, R. Izaks, P. Lago, and H. van Vliet, "A just-in-time architectural knowledge sharing portal," 7th Working IEEE/IFIP Conference on Software Architecture (WICSA 2008), 2008, pp. 125-134, doi:10.1109/WICSA.2008.20.

[19] H. R. de Faria and G. Adler, "Architecture-centric global software processes," Intl. Conf. on Global Software Engineering (ICGSE '06), 2006, pp. 241-242.

[20] J. D. Herbsleb, "Global software engineering: the future of socio-technical coordination," Future of Software Engineering (FOSE '07), 2007, pp. 188-198.

[21] J. D. Herbsleb and R. E. Grinter, "Architectures, coordination, and distance: Conway's law and beyond," IEEE Software, vol. 16, no. 5, 1999, pp. 63-70.

[22] J. D. Herbsleb and A. Mockus, "An empirical study of speed and communication in globally distributed software development," IEEE Trans. on Software Engineering, vol. 29, no. 6, 2003, pp. 481-494

[23] J. D. Herbsleb and D. Moitra, "Global software development," IEEE Software, vol. 18, no. 2, 2001, pp. 16-20.

[24] B. Kitchenham, "Procedures for performing systematic reviews," Joint Technical Report. Keele University and National ICT Australia Ltd., 2004, pp. 1-28.

[25] S. Krishna, S. Sahay, and G. Walsham, "Managing cross-cultural issues in global software outsourcing," Comm. of the ACM, vol. 47, no. 4, 2004, pp. 62-66.

[26] J. A. Laredo and R. Ranjan, "Continuous improvement through iterative development in a multi-geography environment," IEEE Intl. Conf. on Global Software Engineering (ICGSE '08), 2008, pp. 232-236.

[27] J. Noll, S. Beecham, and I. Richardson, "Global software development and collaboration: barriers and solutions," Inroads - SIGCSE Bulletin, in press.

[28] I. Richardson, V. Casey, S. Beecham, J. Burton, and F. Mc-Caffery, "Managing virtual software development teams: a process framework," unpublished.

[29] I. Richardson, V. Casey, J. Burton, and F. McCaffery, "Global software engineering: a software process approach," Collaborative Software Engineering, I. Mistrík, J. Grundy, A. van der Hoek, and J. Whitehead, Eds., Springer-Verlag/Computer Science Editorial, 2010.

[30] I. Richardson, M. O'Riordan, V. Casey, B. Meehan, and I. Mistrik, "Knowledge management in the global software engineering environment," KNOWING: Knowledge Engineering in Global Software Development Workshop, 2009.

[31] A. F. Rutkowski, D. R. Vogel, M. van Genuchten, T. M. A. Bemelmans, and M. Favier, "E-collaboration: the reality of virtuality," IEEE Trans. on Professional Communication, vol. 45, no. 4, 2002, pp. 219-230

[32] F. Salger, "Software architecture evaluation in global software development projects," OTM 2009 Workshops, LNCS 5872, R. Meersman, P. Herrero, and T. Dillon, Eds., Springer, 2009, pp. 391-400.