



# The effect of governance on global software development: An empirical research in transactive memory systems



Christina Manteli\*, Bart van den Hooff, Hans van Vliet

VU University, de Boelelaan, 1081 Amsterdam, The Netherlands

## ARTICLE INFO

### Article history:

Received 10 October 2013  
 Received in revised form 10 April 2014  
 Accepted 10 April 2014  
 Available online 19 April 2014

### Keywords:

Global software development  
 Distributed teams  
 Software governance  
 Transactive memory systems  
 Social network analysis

## ABSTRACT

**Context:** The way global software development (GSD) activities are managed impacts knowledge transactions between team members. The first is captured in governance decisions, and the latter in a transactive memory system (TMS), a shared cognitive system for encoding, storing and retrieving knowledge between members of a group.

**Objective:** We seek to identify how different governance decisions (such as business strategy, team configuration, task allocation) affect the structure of transactive memory systems as well as the processes developed within those systems.

**Method:** We use both a quantitative and a qualitative approach. We collect quantitative data through an online survey to identify transactive memory systems. We analyze transactive memory structures using social network analysis techniques and we build a latent variable model to measure transactive memory processes. We further support and triangulate our results by means of interviews, which also help us examine the GSD governance modes of the participating projects. We analyze governance modes, as set of decisions based on three aspects; business strategy, team structure and composition, and task allocation.

**Results:** Our results suggest that different governance decisions have a different impact on transactive memory systems. Offshore insourcing as a business strategy, for instance, creates tightly-connected clusters, which in turn leads to better developed transactive memory processes. We also find that within the composition and structure of GSD teams, there are boundary spanners (formal or informal) who have a better overview of the network's activities and become central members within their network. An interesting mapping between task allocation and the composition of the network core suggests that the way tasks are allocated among distributed teams is an indicator of where expertise resides.

**Conclusion:** We present an analytical method to examine GSD governance decisions and their effect on transactive memory systems. Our method can be used from both practitioners and researchers as a “cause and effect” tool for improving collaboration of global software teams.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Over the years globalization of software development activities turned into a common practice. Factors such as the coordination and synchronization of the activities across locations and different time zones, the communication and the knowledge management between distributed teams became familiar among scholars and practitioners. And while research in global software development (GSD) evolves and new practices emerge [1], Herbsleb [2] notes that “there is little reason to expect that these factors will diminish”.

The purpose of this paper is not to try to diminish those factors influencing GSD collaboration, but rather identify them and use them as a tool for a “cause and effect” analysis. Particularly, we are interested in investigating how different decisions that companies take on how to govern their GSD activities, affect knowledge management processes, and more specifically the development of transactive memory systems (TMSs). In the following paragraphs, we elaborate on that purpose.

With the continuous and evolving strategies in global software development, there is a turn of interest towards the challenges and the key issues of managing GSD activities [3–5]. As a result, global software development governance turned into an emerging field of research, as a subfield of information technology (IT) governance. The purpose of GSD governance is to identify those aspects that

\* Corresponding author. Tel.: +31 20 59 87 788.

E-mail address: [c.manteli@vu.nl](mailto:c.manteli@vu.nl) (C. Manteli).

are necessary for an effective coordination and collaboration among distributed teams. As Bannerman suggests [6], “governance is the infrastructure needed to ensure the satisfaction of direct and indirect stakeholders”. For instance, when engaged in global activities one can decide to create a captive center in a remote location, or to make a “client–supplier” contract with an external partner. Managers also need to decide on how to structure their teams, and how to allocate tasks among geographically dispersed members. What parts of the projects will be outsourced to the remote partners, and which parts will remain in site? What kind of responsibilities will be delegated to the offshore partners and how much to delegate? These are all questions that frame the governance structure that a company builds for its global activities.

Furthermore, different working practices, geographic proximity and/or legal barriers between remote offices, influence the development of transactive memory systems (TMSs) [7]. Transactive memory is the kind of memory that the team members develop and which helps them identify “who knows what” within the team [8]. In order for the team members to develop such a memory, they have to engage into various transactions through which expertise knowledge is created, shared and stored. As a result, a cognitive system (transactive memory system) is created, where members are aware of each others expertise domain and they are able to access it, update it, share it and facilitate its storage.

Coming back to the main purpose of this study, we pursue the following research question: *How do GSD governance decisions affect transactive memory systems?* We examine four case studies, in two multi-national companies, and we identify their governance structure. Based on the different governance decisions that each case study employs, we report on the differences in the development of transactive memory systems. We present those differences as a “cause and effect” analysis, in order to explain how different GSD *governance modes* (a set of governance decisions) affect the collaboration and communication of the distributed teams.

## 2. Global software development governance

Research on software development governance is rather recent, and as Dubinsky et al. note [9] it is also the result of an increased focus on the human aspects of software development, such as team work and social collaboration. When the software development activities are globally distributed, the need for a clearly defined governance model increases [10]. For instance, the authors in [11] highlight two “moments” of governance in outsourcing relationships; the moment of the promissory contract which is the formal outsourcing contract between remote partners, and a second moment of the psychological contract. The latter involves the post-contractual relationship management that describes the social interactions between the partners such as trust, communication and collaboration aspects.

Several other proposals have been made to define the attributes of a software governance model for distributed development projects, and the coordination mechanisms that such a model should embrace. For instance, Ramasubbu and Balan [12] present a research model on how to create a governance framework for distributed software development, focusing on three stages of a project lifecycle; planning, execution and reflection. In another study, Gewalt and Helbig [10] suggest a governance framework for managing outsourcing engagements based on organizational structures, joint processes and relationship management functions.

In an earlier work [7], we elaborated on global software development governance and we proposed a GSD governance model, based on experiences from an empirical case study. Our research suggested that several knowledge management challenges emerge in GSD settings, caused by the decisions companies take on how to

govern their global activities, on a strategic, tactical and operational level. Building upon the results, we synthesized a multi-site software governance model, based on three aspects: the *business strategy* that binds the mutual relationship of the remote offices, the *structure and composition of the remote teams* and the *way tasks are allocated* across sites. In the present research, we benefit from this model, and we use it to analyze the governance structures of the current case studies.

### 2.1. Business strategy

Carmel and Tija [13] note that one of the things that companies should not forget when they operate in a global environment is the broader strategic goals and their legal implications. Recently, a taxonomy was proposed by Smite et al. [14] in order to categorize and map different GSD strategies. According to that taxonomy, GSD business strategies are identified based on the location (onshore or offshore), the legal entity (insource or outsource), and the geographical distance (Near/Far, Close/Distant). To date, two main business strategies are commonly found in GSD: *offshore outsourcing* and *offshore insourcing* [15]. Offshore outsourcing is when the client-company works with an external partner in a remote location (sometimes referred to as the “buy” decision). Offshore insourcing occurs when the company builds a captive center in a remote location, such as remote offices belong to the same legal entity (also referred to as the “build” decision).

### 2.2. Team structure and composition

Previous research suggests that team structure and composition is a critical factor of good performance in software development [16]. Team size, role descriptions and role distribution are among those characteristics in distributed teams that can influence team coordination and communication and therefore team performance [17,18]. A prominent role in the configuration of GSD team is that of “boundary spanners” [19]. This role is described in various terms such as “brokers”, “bridges”, “liaisons” and “gatekeepers”. Boundary spanning is perceived as a good coordination strategy in the management of distributed collaboration, facilitating team awareness and knowledge management [20]. Boundary spanners are also important within a network because they cover the structural holes that might exist between people, or between sub-groups (clusters). Global software collaboration is, by its nature, a situation where structural holes may emerge between groups that are geographically, temporally and cultural distant [19]. Chang and Ehrlich [21] conclude that “individuals who are more central can exert more influence by virtue of being connected with other powerful individuals in the network, and have access to more resources than less central counterparts”. Finally, Johri [22] recognized two types of boundary spanners within organizations; the *informal boundary spanners*, i.e. people that emerge as mediators between locations, and *formal boundary spanners*, i.e. people designated to play that role.

### 2.3. Task allocation

Several criteria exist on how to distribute work across sites, such as based on the area of expertise, on software architecture and design, or even based on the development steps from requirements elicitation to maintenance [23]. For instance, a common practice for companies that just started outsourcing to a new external partner is to delegate in the beginning only small parts of the development work. This tactic is sometimes referred to also as “throwing requirements over the fence”. As the relationship between distributed offices matures, the client might decide to share more responsibilities with the offshore teams, and work

Download English Version:

<https://daneshyari.com/en/article/550202>

Download Persian Version:

<https://daneshyari.com/article/550202>

[Daneshyari.com](https://daneshyari.com)