Social Capital and Knowledge Networks of Software Developers: A Case Study

VenuGopal Balijepally, Oakland University, Rochester, USA
Sridhar Nerur, University of Texas at Arlington, Arlington, USA

ABSTRACT

Software development is a problem-solving activity, where ideas are combined in complex ways to create a software product that embodies new knowledge. In this endeavor, software developers constantly look for actionable knowledge to help solve the problem at hand. While knowledge management efforts in the software development domain traditionally involved technical initiatives such as knowledge repositories, experience factories, and lessons-to-learn databases, there is a growing appreciation in the software community of the role of developers’ personal knowledge networks in software development. However, research is scarce on the nature of these networks, the knowledge resources accessed from these networks, and the differences, if any, between developers of different experience levels. This research seeks to fill this void. Based on a case study in a software development organization, this research explores the nature of knowledge networks of developers from a social capital perspective. Specifically, it examines the structural and relational dimensions of developers’ knowledge networks, identifies the specific actionable knowledge resources accessed from these networks, and explores how entry-level and more experienced developers differ along these dimensions. The findings from the qualitative analysis, backed by limited quantitative analysis of the case study data underpin the discussion, implications for practice and future research directions.

KEYWORDS

Actionable Knowledge, Knowledge Management, Knowledge Networks, Social Capital, Software Developers, Software Development

INTRODUCTION

The software development field has experienced unprecedented growth in the last over a decade. The imperative for business agility as well as rapid advances in technology have shortened product lifecycle times across organizations (Baskerville & Pries-Heje, 2004). Software development, once believed to be the exclusive preserve of computer programmers, who worked in relative isolation, is increasingly becoming a socio-technical endeavor (Doherty & King, 2005; Luna-Reyes et al., 2005), characterized by extensive collaboration and knowledge sharing. This is particularly manifest in agile development methodologies that emphasize collaborative development through self-organizing teams (Beck & Andres, 2005; Boehm & Turner, 2005; Cockburn, 2000; Batra et al., 2011; Erickson et al., 2005). Members of such teams have to be versatile, capable of working on all aspects of software development and playing roles outside their functional expertise. As Ambler (2004) points out, agile developers have to be “generalizing specialists”. The perceptible shift towards lean principles (i.e.,

DOI: 10.4018/JDM.2019100103

Copyright © 2019, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
eliminating waste, building quality into the process, creating knowledge, quickly delivering value, etc.), as articulated by Poppendieck and Poppendieck (2006), increasingly requires developers to have an expanded skill-set that balances soft and hard skills (Gallagher et al., 2010). In short, the imperative to deliver value in ever-decreasing cycle times compels developers to actively seek information and actionable knowledge during the software development process.

Software development is an inherently complex process that is fraught with uncertainties, both in terms of volatile requirements and technological intricacies (Assimakopoulos & Yan, 2006; Nerur & Balijepally, 2007). Knowledge plays a critical role in mitigating the ambiguity of this process and arriving at an acceptable solution. In this knowledge creation endeavor (Aurum et al., 2003; Bjørnson & Dingsøyr, 2008), software developers are constantly looking for help, in terms of actionable knowledge, to wrap their heads around the problem at hand. It is not uncommon for software developers to scour the Internet—blogs, technical forums and list serves, among others—to acquire problem-specific knowledge (Assimakopoulos & Yan, 2006). Knowledge management (KM) systems, where available, could serve as valuable sources of information relating to business processes, or the technology domain. Appreciating the criticality of knowledge exchange in software development, organizations have been undertaking KM initiatives (e.g., Basili et al., 1994; Desouza et al., 2006; Dingsøyr, 2005; Komi-Sirvio et al., 2002; Rus & Lindvall, 2002).

Despite the proliferation of KM systems and knowledge depositories in IT organizations, there is evidence to suggest that software developers rely to a greater extent on their personal contacts for actionable knowledge and tips (Newell, 2004). For most problems, especially those that embody tacit knowledge, interacting with peers at a personal level appears to be a viable option because it entails less personal cost to developers, relative to other avenues (Desouza, 2003a). Therefore, software developers do look beyond impersonal sources such as the Internet or knowledge repositories for help. Clearly, this is not consistent with the stereotypical perception of software developers as nerds who relish and excel on their one-on-one dealings with computers, rather than being adept at social interactions with people (Fitz-Enz, 1978). In this regard, there is some evidence of research interest in the knowledge networks of software developers (e.g. (Assimakopoulos & Yan, 2006; Aurum et al., 2008; Desouza, 2003; Méndez-Durón & García, 2009). Also, there is some research to understand the differences in the knowledge seeking behaviors of developers of varying experience levels (e.g. Desouza et al., 2006; Walz et al., 1993). However, IS research on the knowledge networks of software developers is still in its infancy.

Some of the findings in management literature relating to the actionable knowledge afforded by social capital to individuals in organizations is based on study of knowledge workers in consulting teams (Cross & Sproull, 2004). We believe, unlike other types of knowledge workers, software developers may need to access actionable knowledge much more expeditiously, due to some unique characteristics of software development task. For instance, developers in agile teams are expected to analyze requirements, code and release working software regularly which requires collaborating on a continuous basis with the end users/product owners (Batra et al., 2011). Practices such as daily stand-up meetings, developer role rotation, quick prototyping all require that developers are able to access any actionable knowledge that they need at any stage of their daily work expeditiously to stay productive. So, developers certainly have a great incentive to seek such actionable knowledge in a highly efficient manner so as not to get bogged down. Knowledge ambiguity in software development tasks, that results from the specificity, complexity and tacitness of knowledge, could certainly be high due to the diverse forms of knowledge involved (e.g., domain knowledge, technical knowledge, project scheduling knowledge, etc.). Empirical research suggests that knowledge ambiguity adversely affects knowledge transfer (Simonin, 1999; Van Wijk et al., 2008). So, it is reasonable to expect some differences in knowledge seeking behaviors between software developers and other knowledge workers in organizations based on the nature of their work.

There is, as yet, no clear understanding of the personal networks of developers. Neither do we fathom the differences, if any, between the networks of developers of different experience levels.
Understanding Functional Dependency
www.igi-global.com/chapter/understanding-functional-dependency/9216?camid=4v1a

Self-Tuning Database Management Systems
www.igi-global.com/chapter/self-tuning-database-management-systems/20761?camid=4v1a

Data Modeling in UML and ORM: A Comparison
Terry Halpin and Anthony Bloesch (1999). *Journal of Database Management* (pp. 4-13).
www.igi-global.com/article/data-modeling-uml-orm/51222?camid=4v1a