Chapter 64
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.

DOI: 10.4018/978-1-6684-3702-5.ch064

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., 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.