Expert Systems with Applications 41 (2014) 126–136

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Organizational learning networks that can increase the productivity of IT consulting companies. A case study for ERP consultants

Razvan Bologa *, Ana Ramona Lupu

Computer Science Department, Academy of Economic Studies, Piata Romana 6, Bucharest, Romania

ARTICLE INFO

Keywords:
Learning social networks
Knowledge sharing
Software development education
Enterprise resource planning
ABAP

ABSTRACT

This paper considers the use of social learning networks to increase the productivity of IT consulting companies. We advocate that using a carefully designed social learning network can reduce the learning time for enterprise software developers and consultants. By viewing learning as a social act, a consulting company can increase its productivity. Increased productivity is based on hastening the learning process. The focus of this paper is to identify the ways in which social networks catalyze the process of knowledge sharing in order to increase the productivity in the enterprise resource planning (ERP) consulting sector. We present a set of detailed practical results that were obtained from experiments with an original knowledge sharing method that was applied to training young software developers to enable them to work for some of the world's most demanding IT companies. The experimental data were collected from 2004 to 2011 during 12 training sessions conducted by an IBM partner in conjunction with the Computer Science Department of a large Eastern European University. The main results of this study were: (1) designed a learning community that reduced the time needed to insert junior consultants into ERP projects; and (2) statistical data were generated that measured the increase in productivity that an ERP consulting company could obtain by employing organizational learning networks. We also discuss the positive impacts of social networks that can be established between private companies and universities.

1. Introduction

There has been substantial research in the area of organizational learning. By definition, organizational learning is the acquisition of information knowledge and skills by individuals (Argyris & Schon, 1995; Easterby-Smith & Lyles, 2011). Organizational learning networks have been previously studied with regard to their relationships with the productivity of IT consulting companies. Skerlavaj, Dimovski, Mrvar, and Parhor (2010) found that social learning networks had a positive impact inside IT consulting companies, including those involved in enterprise resource planning (ERP) consulting.

One of the most prominent researchers in this field claimed that the only sustainable competitive advantage for a company was its ability to learn faster than its competitors (Geus, 1988; Kumar, Jones, Venkatesan, & Leone, 2011). Learning is crucial for an IT consulting company because it only sells knowledge and, thus, it has to be better than its competitors.

A consulting company's productivity is not always easily defined. Nachum (1999) observed that productivity models developed in the manufacturing industry did not apply to professional companies. These models depend on the particular type of service that is being offered.

The main issue that an IT consulting company must solve when it comes to its productivity is the considerable time needed to insert a consultant into a project that is unknown to that consultant. In terms of ERP software packages, the time that a person must spend learning before (s)he can start working autonomously is much longer that in other industries.

According to Lazowska (2011), who cites a report from the US Bureau of Labor Statistics, we are currently facing a shortage of computer specialists that will continue to grow until at least 2018. In the US alone, from 2008 to 2018 the job market will be in demand for an extra 1.4 million new computer specialists. This is not a new situation, as it has been occurring for many years. The US President's Council of Advisors on Science & Technology stated in December 2010 that "all indicators – all historical data, and all projections – argue that (computer science) is the dominant factor in America's science and technology employment" (Lazowska, 2011).

This shortage of skills also applies to ERP consulting companies, as ERP has been a hot sector since the mid 1990s. This paper was written based on the assumption that the productivity of a consulting company in an area with a shortage of resources is directly

* Corresponding author.
E-mail addresses: razvanbologa@ase.ro (R. Bologa), ramona.lupu@ie.ase.ro (A.R. Lupu).
URL: http://www.ase.ro (R. Bologa).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.07.016

proportional to the number of new junior consultants that it manages to successfully insert into projects over a certain period of time. Reducing the time needed to train a junior consultant in order to become stand-alone is crucial for increasing a consulting company's productivity. Additional details regarding the interpretation of productivity and statistical data that confirm this assumption are provided in the following chapters.

This paper has four main parts: (1) background, in which the problem is formulated in detail; (2) the learning model and the proposed social learning network; (3) the learning process that was conducted and the statistical data that were obtained; and (4) a discussion on productivity gains and answers to some questions and criticisms.

2. The need for faster training

The various aspects of knowledge creation are critical for sustaining the development of learning organizations (Easterby-Smith & Lyles, 2011; Selamat & Choudrie, 2007). In the IT industry, the competitiveness of a company is largely determined by the knowledge it possesses, and the knowledge of an organization is considered to be derived from its employees (Lin, 2006; Skyrme, 2012; Von Krogh, Ichijo, & Nonaka, 2000).

Due to a shortage of skills (Lazowska, 2011), providing the right training in a minimum amount of time is a critical factor for any software company that is active in today's highly competitive IT industry. This is why many companies are now looking for ways to reduce their employees' training periods and to train them in a manner that is faster than conventional on-the-job training.

To accelerate the educational process, we have developed a social framework for accelerating the sharing of knowledge between experienced software developers and trainees inside a social network. This framework consists mainly of creating a social learning network that catalyzes the knowledge sharing process and increases learning speed. This model began with the bold idea of an Italian IBM partner company that approached an Eastern Europe university in 2004. This company proposed the development of a method that would transform young IT graduates into internationally competitive ABAP programmers in six-eight months instead of the industry's standard time of two years. The endeavor that followed was successful, and the result was a framework that ensured both the transfer of explicit and tacit knowledge.

When the Italian entrepreneur launched the idea of reducing the time for educating ABAP developers from two years to six-eight months, it created mixed reactions. ABAP is the programming language for SAP, the world's largest and most sophisticated ERP solution. Since the early 1990s, most large multi-national companies have adopted SAP as an integrated information system, which has generated a huge need for skilled technical consultants (Bjorlin, 2008). ABAP has been used for over 30 years to write the code for SAP. Although it is very simple as a programming language, working in ABAP requires a thorough understanding of the way in which SAP was written over the last three decades.

Learning ABAP requires learning a vast amount of detail about SAP's database structure, the various tools used in SAP, and the way it interacts with other applications. A certain level of understanding of business procedures is also a must, as ABAP programmers often need to work in situations with fragmented specifications. In addition, because SAP is used in key processes inside large multi-national companies, the error tolerance is very low. Basically, an ABAP application that handles billions of Euros or dollars needs to work perfectly.

For these reasons, it takes considerable time to train a new university graduate in computer science to become an ABAP developer who is capable of working autonomously. By 2004, major companies like IBM considered that the standard time for fully training an ABAP developer was 24 months. There is also a standard SAP curriculum that is used in this training process. By the time that IBM's partner had approached us, this was the standard training time accepted by the industry.

The demand for ABAP developers has always been greater than the supply (Bjorlin, 2008). This was also confirmed by the IBM managers that backed this project. This was the main reason why IBM's partner company financed this research project; they needed to reduce their training time. We found that, even after the economic crisis that began in 2008, the demand for ABAP developers remains very high.

The framework for learning developed in the project described in this paper was not limited to ABAP technology. It was actually used for other technologies that presented similar difficulties when being learned by young university graduates, such as Business Intelligence (Hyperion). Additional details regarding these technologies and the results will be described later. More and more software technologies in the IT sector are being widely adopted, which is creating a shortage in the supply of skilled consultants.

3. Knowledge management considerations

After initial discussions, we decided to become involved in the research project proposed by IBM's partner. The idea of the primary author of this paper was to use the research of Everett Rogers on the diffusion of innovations. In 1962, Rogers published a very well known sociology book entitled *Diffusion of Innovations*, which analyzed why some communities adopted innovations faster than others. This was combined with ideas from more recent research on organizational learning and how it can be used to spread knowledge inside a community (Easterby-Smith, Crossan, & Nicolini, 2000; Skyrme, 2012).

The reason for starting with the research of Rogers was that his ideas were considered to be the closest to the concept of learning innovation faster. By their nature, software technologies are highly innovative. It is quite common to see a considerable amount of innovation generated by software companies each year. We assumed that learning a sophisticated technology by a group of students was, from a sociological point of view, the dissemination of innovation at the group level.

The theory of the "diffusion of innovations" used as a background for our research proposes four main elements that influence the spread of a new idea: the innovation; communication channels; time; and a social system (Rogers, 1995).

While analyzing Rogers' theory to reduce the time needed for training software developers, we observed that, in order to reduce the time, it was necessary to improve another element among these four main elements. Essentially, if the time was to be shortened, something else had to be increased.

The first element, the innovation, could not be changed because it was fixed. The innovation that had to be disseminated in our case was the SAP and ORACLE technologies, which were already running in many companies; thus, nothing could be done to change this.

The second element, communication channels, was also something that we could not see ways to improve. The communication channels used in the IT industry are based on the most advanced technologies; thus, we decided not to improve the communication channels.

The only element that could be manipulated to reduce the time was the last: the social system. The social system was relatively unexplored in the companies with which we dealt at the beginning of this research study. In 2004, the area of organizational learning was a research field in its infancy (Easterby-Smith et al., 2000). In software companies, most of the social set-ups of teams were left