Chapter 1
Research Review: Globally Distributed Requirements Engineering and Agility

Vanita Yadav
Institute of Rural Management Anand (IRMA), India

ABSTRACT

Getting requirements right is considered the most important and difficult part of the software development process. As organizations and stakeholders become more globally distributed, getting requirements right will pose a greater challenge. Today, organizations are offered a sophisticated array of multimedia meeting systems with video, audio and computer support for remotely specifying requirements. In the context of present day's global workforce scenario, distributed requirements engineering is now being explored by researchers as well as industry to build an understanding of the dynamics of this highly interactive phase in a globally distributed context. Accordingly, this paper addresses the timely need of reviewing the literature on globally distributed requirements engineering. Findings highlight paucity of literature in this area. Additionally, the paper presents an innovative approach to globally distributed requirements engineering by reviewing the literature on the applicability of agile development approach in a globally distributed setting. Contributing to academic and practitioner literature, the author discusses emerging directions for future research in this area.

INTRODUCTION

Most organizations use a standard set of steps, called the systems development methodology to plan, analyze, design, implement and maintain their information systems (Hoffer, George, & Valacich, 2005). One of the hardest parts of system's development is deciding *what the system should do*, that is in determining the system requirements (Crowston & Kammerer, 1998, p. 227). In his classic essay "No silver Bullet", Frederick Brooks (Brooks, 1975) noted that:

The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all interfaces to people, to machines

DOI: 10.4018/978-1-4666-2934-9.ch001

and to other software systems. No part of the work cripples the resulting system if done wrong. No other part is more difficult to rectify later.

Thus getting the requirements right may be the single most important and difficult part of the software development process (Guinan, Cooprider, & Faraj, 1998). The key result of requirements problems is rework that implies redoing something that was already complete. Rework can consume 30 to 50 percent of total development cost (B. W. Boehm & Papaccio, 1988), and requirements errors account for 70 to 85 percent of the rework cost (Leffingwell, 1997). Figure 1 illustrates that it costs much more to correct a defect which is found late in a project than to fix it shortly after its creation (Grady, 1999; King & Marasco, 2008). Weigers (2003) asserts shortcomings in requirements practices can pose a great risk to a project success.

DISTRIBUTED REQUIREMENTS ENGINEERING

As organizations become more 'virtual', distributed development will become more apparent throughout the entire lifecycle, including analysis of software requirements in the early stages of the development lifecycle (Evaristo, Watson-Manheim, & Audy, 2005). Ocker et al. (1995) reported the successful usage of computer conferencing to support groups working on determining software requirements in an asynchronous distributed environment.

As mentioned earlier, getting requirements right is a software project's most important part (Hofmann & Lehner, 2001) and its success is critical for achieving project success (Bhat, Gupta, & Murthy, 2006). Despite the abundant literature available on globally distributed virtual teams (Powell, Piccoli, & Ives, 2004) and outsourcing (Dibbern, Goles, Hirschheim, & Jayatilaka, 2004; Yadav & Gupta, 2008), there are very few studies addressing requirements engineering activities like requirements analysis in distributed software development (Yadav, Nath, Adya, & Sridhar, 2007). Requirements analysis involves analysts working with end users/clients to understand the organizational information processing needs and develop IS objectives. Based on the gathered requirements, the analyst builds a model representing 'what' the customer requires which establishes a basis for creation of software design.

Nowadays, organizations are offered a sophisticated array of multimedia meeting systems with video, audio and computer support for remotely specifying requirements (D. E. H. Damian, Eberlein, Shaw, & Gaines, 2000). Many researchers have argued that groupware technology has supported distributed requirements engineering and modeling perspectives (D. E. H. Damian et al., 2000; Robinson, 1990). IS researchers have suggested that development of information systems in general, and the upstream portions e.g. defining requirements, of the development process in particular could benefit from creative and innovative solutions (Couger, Higgins, & McIntyre, 1993; Ocker et al., 1995). One such approach suggested by researchers is usage of group support systems (Dubrovsky, Kiesler, & Sethna, 1991; Jarvenpaa, Rao, & Huber, 1988). Ocker et al. (1995) reported the successful usage of computer conferencing to support groups working on requirements determination in an asynchronous distributed environment.

Figure 1. Variations in cost to fix a software defect