Information and Organization 20 (2010) 44–63

Contents lists available at ScienceDirect

Information and Organization

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

Getting on the same page: Collective hermeneutics in a systems development team

Sean Hansen [a,*], Julie Rennecker [b]

[a] Weatherhead School of Management, Case Western Reserve University, Cleveland, OH 44106, USA
[b] Panoramic Perspectives, P.O. Box 26263, Austin, TX 78755, USA

ARTICLE INFO

Article history:
Received 19 March 2008
Received in revised form 7 January 2010
Accepted 7 January 2010

Keywords:
Information systems development
Hermeneutics
Collective meaning-making
Interpretive processes
Organizational discourse
Interpretive episodes

ABSTRACT

The inherent complexity of information systems development presents significant impediments to the achievement of shared meaning among the members of a development team. How then do software development teams resolve questions of shared meaning in the development process? In this study, we build upon observations of a large platform development team to identify the ways in which team members converge around shared meanings through the application of a repertoire of interpretive techniques. Specifically, we develop a model of interpretive team interaction. This collective hermeneutic model extends the hermeneutic tradition in IS research by addressing the ways in which an interpretation takes shape not simply within the mind of an individual but also through collaboration with others. Finally, we discuss implications of this theoretical perspective for the design of systems development environments and the prospect for additional research on the interpretive processes of development teams.

1. Introduction

The development of large information systems platforms and applications represents one of the most complex undertakings in contemporary business. The significant scale of development efforts and the uncertainty of design efforts combine with the diverse nature of software development teams to create tremendous coordination demands (Espinosa et al., 2001; Kraut & Streeter, 1995; Tellioglu & Wagner, 1999). These coordination challenges are further augmented by the increasing geographic

* Corresponding author. Tel.: +1 216 368 2114; fax: +1 216 368 4776.
E-mail addresses: hansen@case.edu (S. Hansen), julie@panoramicperspectives.com (J. Rennecker).

1471-7727/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.infoandorg.2010.01.001

and temporal distribution of development team members (Herbsleb & Mockus, 2003; Im, Yates, & Orlikowski, 2005).

Not surprisingly, software development has been repeatedly described as an inherently social process (Faraj & Sproull, 2000; Hirschheim, Klein, & Lyytinen, 1995), rather than a merely technical one. The development team for a large information system requires representation from a range of specialty skill domains, including project managers, analysts, programmers, and network support personnel. Indeed, for very large initiatives, multiple teams are engaged, each focused on a particular aspect of the final system. In addition, development teams must interact with myriad other organizational members as they design, develop, and implement software resources. With each layer of organizational complexity, the challenges of interpersonal and group dynamics are increased (Curtis, Krasner, & Iscoe, 1988; Roberts, Cheney, Sweeney, & Hightower, 2004).

Development teams employ a wide range of formal processes and technical artifacts to manage their coordination demands (Im et al., 2005; Tellioglu & Wagner, 1999). Such resources address the interdependence of development tasks and ensure that most of the development process can proceed without explicit debate over the collective understanding of task objectives. However, in a development process marked by inherent heterogeneity, questions of shared *meaning* are inevitable. When such circumstances arise, how do software development teams resolve questions of shared meaning in the development process?

The current study explores the processes that software development teams employ to arrive at common understandings in the day-to-day conduct of their work. Specifically, we propose that development teams engage in a collective hermeneutic process to explore the variety of interpretations that exist around a given topic and to accomplish collective meaning-making so that the development process can proceed. Broadly understood, *hermeneutics* refers to the science (or art) of interpretation. While it has its origins in ancient thought, hermeneutics became the source of significant philosophical inquiry in the 20th century (Grondin, 1991). We argue that the process of meaning-making within development teams represents a collective approach to the interpretation of "texts" encountered in the design environment. Drawing upon the work of Ricoeur (1979, 1991), we understand a *text* to mean something that must be interpreted. As such, a text may include traditional written artifacts as well as utterances or actions intended to convey some meaning within a given context. Similarly, we take *discourse* to mean a language event – i.e., the message produced through the use of language (Ricoeur, 1979). Taken together, these definitions suggest that a text is the embodiment of a discourse. Building upon an observational study of a software development team on a large platform project, we develop a model of the hermeneutic process that teams employ in the pursuit of collective meaning.

In Section 2, we explore the context of software development and some of the challenges of achieving collective understanding. This is followed in Section 3 by a discussion of the literature on group meaning-making and the theoretical foundations for the model developed later in the paper. Section 4 presents the research effort conducted through observation of a large international software development team. Section 5 presents the study findings and the resulting model of a collective hermeneutic process. Section 6 offers a discussion of the key insights from the study and their implications for both the software development process and the study of information systems development. The study concludes with an acknowledgement of the limitations of the study and suggestions for further research.

2. Systems development in teams

It has been widely observed that systems development efforts are exceedingly complex undertakings due to a combination of technical complexity and the organizational challenges inherent in the collaboration of multiple stakeholder groups spanning a wide range of functional domains (Ewusi-Mensah, 1997; Roberts et al., 2004). A central source of project complexity is the essentially social nature of the systems development process (Faraj & Sproull, 2000; Sawyer & Guinan, 1998). The vast majority of large scale systems development efforts are executed through a team structure (Guinan, Cooprider, & Faraj, 1998; Reid & Wilson, 2007). This is true of both in-house development efforts and the creation of packaged software (Carmel, 1997). A team format implies significant issues