

4-10-2008

Collective Hermeneutics in a Systems Development Process

Sean Hansen

Case Western Reserve University

Julie Rennecker

Case Western Reserve University

Follow this and additional works at: http://aisel.aisnet.org/sprouts_all

Recommended Citation

Hansen, Sean and Rennecker, Julie, "Collective Hermeneutics in a Systems Development Process" (2008). *All Sprouts Content*. 124.
http://aisel.aisnet.org/sprouts_all/124

This material is brought to you by the Sprouts at AIS Electronic Library (AISeL). It has been accepted for inclusion in All Sprouts Content by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Collective Hermeneutics in a Systems Development Process

Sean Hansen

Case Western Reserve University, USA

Julie Rennecker

Case Western Reserve University, USA

Abstract

The inherent complexity of information systems development presents significant impediments to the achievement of shared meaning among the members of a development team. In addition to the technical requirements of systems development, its intensely social nature challenges project teams to unite around a collective understanding of the processes and objectives that they pursue. 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 a repertoire of interpretive techniques. Specifically, we develop a model of interpretive team interaction that is based on the concept of a collective hermeneutic process. The 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.

Keywords: Information Systems Development, Hermeneutic Processes, Collective Meaning-making

Permanent URL: <http://sprouts.aisnet.org/6-6>

Copyright: [Creative Commons Attribution-Noncommercial-No Derivative Works License](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Reference: Hansen, S., Rennecker, J. (2006). "Collective Hermeneutics in a Systems Development Process," Case Western Reserve University, USA . *Sprouts: Working Papers on Information Systems*, 6(6). <http://sprouts.aisnet.org/6-6>

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 (Kraut & Streeter, 1995; Tellioglu & Wagner, 1999). Not surprisingly, software development has been repeatedly described as an inherently social process (Hirschheim, Klein, & Lyytinen, 1995; Sawyer, Farber, & Spillers, 1996). 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 a myriad of other organizational members as they design, develop, and implement software resources. With each additional layer of organizational complexity, the challenges of interpersonal and group dynamics are increased (Curtis, Krasner, & Iscoe, 1988). As a result of this social heterogeneity, development efforts are characterized by significant interdependence (Crowston & Kammerer, 1998; Kraut & Streeter, 1995).

Development teams employ a wide range of formal processes and technical artifacts to address their coordination demands (Tellioglu & Wagner, 1999). Such resources address the interdependence of development activities 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; Tice & Slavens, 1983). 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. Building upon an observational study of a software development team engaged on a large platform project, we develop a model of the hermeneutic process that teams employ in the pursuit of collective meaning.

Section 2 of the study explores the context of software development and some of the challenges to the achievement of collective understanding. This is followed by a discussion of the literature on group meaning-making and the theoretical foundations for the model developed in the study. 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 the software development process and the study of information systems development. The study concludes with an acknowledgement of limitations and suggestions for further research.

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 (Brooks, 1987; Ewusi-Mensah, 1997; Sawyer & Guinan, 1998). A central source of project complexity is the essentially social nature of the systems development process (Hirschheim et al., 1995; Sawyer et al., 1996; Sawyer & Guinan, 1998). Nearly all systems development efforts are executed through a team structure (Guinan, Coopriider, & Faraj, 1998). This is true of both in-house development efforts and the creation of packaged software (Carmel, 1997). A team format implies significant issues regarding group dynamics, including the concerns of goal setting, group cohesion, and interpersonal conflict (Ewusi-Mensah, 1997; Guinan et al., 1998). The questions of group dynamics are often exacerbated by other characteristics of a project approach. The time-bounded nature of project work limits the allowances for resolving interpersonal or political issues prior to the commencement of productive work. Similarly, systems development projects are capital-intensive, placing significant pressure on team efficiency and cost management.

One of the most important characteristics of systems development project teams is the diversity of knowledge that they require (Curtis et al., 1988; Levesque, Wilson, & Wholey, 2001; Levina & Vaast, 2005; Tellioglu & Wagner, 1999; Walz, Elam, & Curtis, 1993). To address the complex organizational problems set before them, project teams bring together individuals from a wide variety of technical and functional domains. The challenges of this heterogeneous knowledge mix become even more acute as project teams interact with other stakeholder groups (Curtis et al., 1988). For instance, most large systems development projects involve the interaction of multiple teams, each engaged on a specific facet of the overall effort. In addition, project teams must work with representatives from other stakeholder groups within the organization, such as executive managers, user groups, and accounting personnel. In many cases, the need for effective interaction extends beyond the walls of the organization to incorporate external customers, corporate partners, and vendors. Thus, the integration of domain knowledge from a broad array of sources is a defining task of any development project.

The demands of knowledge integration imply the need for significant attention to communication and coordination. Indeed, the absence of effective coordination practices has been cited as a primary contributing factor in high rates of information systems failure (Kraut & Streeter, 1995). To avoid breakdowns at multiple stages in the development process, project teams must ensure that they are able to establish shared mental models of the development process and objectives (Curtis et al., 1988; Guinan et al., 1998; Walz et al., 1993). Furthermore, they have to communicate this shared vision to other stakeholder groups. While some evidence suggests that project teams may fail to converge around models of their problem space (Levesque et al., 2001), accepted wisdom holds that “team members need to be *speaking the same language* ... in order to share knowledge about the system” (p. 63, Walz et al., 1993, italics in original). But what is the process by which this common language emerges in the context of a given project? How do members of a project team accomplish collective meaning-making to enable action and a shared vision?

Among the ways that systems development professionals have sought to address these coordination and communication demands is the application of formal techniques and design resources. The software development life cycle (Boehm, 1988; Hirschheim et al., 1995), with its

structured analysis, staged development and deliverables, and emphasis on review and sign-off, reflects an effort to address coordination challenges through a formal method of communication. Variations on this structured analysis method still predominate in systems development practice. Project teams also implement shared tools to ensure a common design environment and platform for project documentation (Tellioglu & Wagner, 1999). Teams may even establish project “war rooms” to improve access to information and awareness of project issues among members of the development team (Sawyer et al., 1996; Teasley, Covi, Krishnan, & Olson, 2002). In addition to formal coordination mechanisms, members of a development team rely heavily on informal communication between individuals (Kraut & Streeter, 1995; Sawyer & Guinan, 1998; Walz et al., 1993). This is especially true during requirements definition and early stages of a project or on projects marked by significant ambiguity. Informal communication between individuals provides the basis for significant learning by team members and serves as an important boundary spanning role between functional units (Kraut & Streeter, 1995).

Team meetings, one of the primary communication mechanisms employed on development projects, combine the elements of formal and informal communication. Meetings represent a formal structure that also creates opportunities for informal interaction among members. Despite the considerable expense associated with them (Levesque et al., 2001), meetings represent the predominant coordination mechanism on most software development projects (Faraj & Sproull, 2000). In fact, some systems developers have estimated that the bulk of their time is spent in project-related meetings with a minor portion of the work day dedicated to technical development (Curtis et al., 1988). Nevertheless, meetings persist because they form a key basis for the collective meaning-making necessitated by a development effort (Sawyer et al., 1996; Sawyer & Guinan, 1998; Walz et al., 1993).

Collective Meaning-Making

The concept of *meaning* itself defies a simply definition, but much of 20th Century social science has pursued it through a focus on “shared or common responses, significations, intentions, and goals, and in general, the interpretive and representational processes that underlie human conduct” (Maines, 2000). The issue of collective meaning-making within organizations has been addressed extensively in the organizational studies discipline and related research communities. For more than four decades, researchers have wrestled with questions of how individuals arrive at understandings of the world that are sufficiently common to support organized action. In a review of approaches to collective sense making, Donnellon, Gray, & Bougon (1986) identify a number of key theoretical threads in the literature, including explanations based on the conceptions of culture and social exchange. The authors then add to this tradition with a theory of organized action built upon a shared repertoire of communication.

Culture

One the oldest approaches to the issue of shared meaning focuses on the role of culture within organizations. Since the 1970s, researchers have drawn lessons from the field of anthropology to explore cultural perspectives on organizational action. However, this tradition has been marked by the absence of a singular understanding of *culture* itself. Smircich (1983) outlines a variety of key approaches to the issue of culture that have emerged in the organization studies and management literatures. The concept of *corporate culture* presents culture as a characteristic of a given organizational context – an internal variable (Chen, Sawyers, &

Williams, 1997; Tichy, 1982). Within this tradition, culture is understood to be “the social or normative glue that holds an organization together” (Smircich, 1983: 344). It is the “personality” of a firm, expressing the shared ideals of organizational members. An alternative approach that adopts a cognitive perspective on culture is dubbed *organizational cognition*. Through this lens, culture is viewed as a system of beliefs and cognitive frameworks that are shared by all members of an organization – “whatever it is one has to know or believe in order to operate in a manner acceptable to its members” (Goodenough, 1964: 36). From this perspective, the concepts of culture and organization are one-and-the-same – networks of individuals with a common frame of reference.

Both of these approaches to organizational culture offer some insights as to why organizational action is possible in the first place; i.e., how individuals are able to work together and communicate. However, they are less applicable for exploring the ways in which an organizational subgroup arrives at a common vision when meaning is not provided by the environment itself. In the case of a systems development effort, team members may be able to interact because of the cultural beliefs that they share, but that leaves significant ambiguity with respect to the specific objectives to be pursued in the project. Such ambiguity must be addressed if the project is to progress effectively.

Organizational Action as Exchange

A second fundamental approach to the determination of collective meaning and action draws heavily on the work of Karl Weick (Daft & Weick, 1984; Weick, 1979). In this perspective, the conventional relationship between meaning and action is inverted – action occurs first, and meaning is ascribed to it retrospectively. Individuals can engage in collective action as long as they agree upon the means that action will require. A common objective may subsequently emerge, but it is not a prerequisite for organized behavior. In this vision of collective action, the primary unit of analysis is the *double interact*. A double interact is a contingent response pattern in which the action of one individual elicits a response from another individual, whose behavior in turn prompts an adjustment by the initial actor. Broader organizational action then can be understood as a complex network of double interacts; collective behavior is an aggregation of dyadic exchanges. In such exchanges, the requirement for shared meaning is again quite minimal. The individual participants in an exchange (i.e., a double interact) need not agree upon objectives as long as they have a common understanding of each party’s actions.

This perspective on organizational action and shared meaning presents an acute challenge to conventional understandings of a systems development project team. We have seen that a shared vision of project outcomes and processes is understood to be the *sine qua non* of an effective development effort, but Weick is suggesting that such a common understanding is far from required. However, it is worth noting that the sense-making model of organizational action depicts cycles of convergence and divergence around organizational means and ends. Thus, it does not imply that shared meanings play no role in collective action, only that they may be more or less critical at various stages in the cyclical interactions of a group. Given the observation that systems development teams explicitly pursue the establishment of a common vision around a project, it is reasonable to assert that the systems development effort represents a critical occurrence of intended convergence in organizational life. Unfortunately, this broader model of organizational sense-making offers us limited insight on the process by which groups pursue and achieve such convergence.

Organizational Action through Equifinality

A third line of inquiry into collective sense making follows from the Weickian argument by suggesting that shared meaning is unnecessary for organized action. Donnellon et al. (1986) argue that diverse groups can engage in collective behavior based on the concept of *equifinality* – a concept from systems theory meaning that multiple routes can lead to the same outcome (Katz & Kahn, 1978). Rather than pursuing shared meaning of a given objective, organization members can act in concert as long as they establish *equifinal meaning*, or differential understandings that have the same behavioral implications. Members of a group could have different reasons for taking some action and even different beliefs about the possible outcome of the actions, but they will act collectively if they agree upon the necessity of the action in question. For Donnellon et al. (1986), such collective action is made possible not through shared meaning, but through a shared repertoire of communicative behaviors, specifically the use of metaphor, logical argument, affect modulation, and linguistic indirection. However, they reached this conclusion from studies of undergraduate students participating in simulated episodes of organizational decision-making, an environment markedly different from that of an active systems development team. Furthermore, how can the model of equifinal meaning be applied if it is observed that development teams actively pursue a common understanding around processes and objectives?

Though limited in their applicability to the workings of development teams, each of these perspectives on collective meaning-making (or absence thereof) shares a common thread in recognizing the interpretive nature of organizational discourse. As noted earlier, meaning is inextricably linked with the social processes from which it emerges. It is through acts of interpretation that this emergence is achieved. Thus, as we explore the basis of meaning-making within teams, a return to a broader, more fundamental perspective may be warranted.

Hermeneutics

Hermeneutics refers to the science of interpretation, or “what occurs when a really or apparently unfamiliar meaning is made intelligible” (Grondin, 1991: 18). While the term itself emerged in the 17th century, the hermeneutic concept has its origins in ancient thought, most notably that of the Greeks. Prior to the 20th century, hermeneutic methods were applied primarily in the exegesis of sacred, classical, or judicial texts. The development of hermeneutics into a formalized method of textual analysis in the modern era came with the work of Friedrich Schleiermacher, whose perspective remained largely theological. Schleiermacher, and Wilhelm Dilthey after him, emphasized the importance of transcending one’s own temporal and environmental context to get at the true meaning of a text. It is one’s own present situation that blocks the valid understanding of a text, and therefore it must be overcome.

In the 20th century, with the emergence of *philosophical hermeneutics*, the emphasis shifted away from capturing the context of the original author to the act of interpretation itself. This development is largely rooted in the thought of Martin Heidegger (1962), who understood interpretation to be a primary mode of human existence (Tice & Slavens, 1983). Heidegger’s insights were subsequently taken up and extended by the philosopher most closely associated with contemporary hermeneutic theory – Hans-Georg Gadamer. One of the primary contributions of Gadamer’s hermeneutics is a thorough rejection of the perspective on the historicity of the interpreter developed by Schleiermacher and Dilthey. For Gadamer, far from being an impediment to a proper understanding, the prejudices of the individual are essential in any understanding: “Prejudices are biases of our openness to the world” (Gadamer, 1976: 9). In

this case, Gadamer is focusing on the classical sense of the term *prejudice* as a pre-understanding or judgment. We cannot escape from prejudices for they lie at the center of our experience of being. The important task then in hermeneutics is to surface (to the degree possible, which is necessarily limited) our prejudices and to consider reflectively the ways in which they impact the meaning of a text for us. As Gadamer notes, “reflection on a given pre-understanding brings before me something that otherwise happens *behind my back*” (Gadamer, 1976: 38, italics in original).

Key insights from the hermeneutic tradition include the concept of the hermeneutic circle and the extension of hermeneutic analysis to the realm of action. The *hermeneutic circle* is essentially a systems-oriented concept whereby an individual engaged in an interpretive effort must move between understanding a part (e.g., a word or phrase) with respect the whole (e.g., an entire text), and understanding the whole by grasping its composite parts. The “circle” implies the continuous movement back and forth between a constitutive element and the text as a whole.¹ The term hermeneutic circle comes from Schleiermacher, but it has been adopted (and adapted) by those who have followed him. For example, Paul Ricoeur (1979) argues that the hermeneutic circle reflects the interpreter’s recursive movement between “guessing” (i.e., positing an interpretation) and “validating” that guess. In addition, by illustrating the parallels between text and action (specifically in the discursive elements that they share), Ricoeur (1979) argues that interpretive principles must be extended beyond a textual focus to all aspects of social interaction.

The applicability of hermeneutic principles to the broader social science discourse has been recognized by several researchers within the IS community. While hermeneutics is acknowledged as a theoretical underpinning of the entire interpretive IS research tradition (Lee, 1991; Myers, 1997), the explicit emphasis on hermeneutics in IS has been largely driven by the work of Richard Boland (1985, 1987; 1991; Boland & Day, 1989). In one widely-cited work, Boland (1985) suggests three levels on which hermeneutic analysis can inform the IS research process: *use*, *design*, and *study* of information systems. The use of an information system is a hermeneutic task in which system output is interpreted by individuals other than the systems developer. Design in turn is a hermeneutic process in which the “text” being interpreted is the organizational action which a system is intended to support. Finally, IS scholars can turn the mirror upon themselves by acknowledging the interpretive nature of their research.

Hermeneutics has also been taken up by other researchers within the IS research community to explicate a range of phenomena. Hirschheim & Klein (1989) and Hirschheim et al. (1995) explore the hermeneutic underpinnings of systems development methodologies and data modeling approaches. Myers (1994) discusses the integration of philosophical hermeneutics with the critical theory of Jurgen Habermas, and proposes that the resulting *critical hermeneutics* can serve as an appropriate theoretical framework for assessing IS implementation efforts. Similarly, Davis et al. (1992) argue for the use of a hermeneutic method in the diagnosis of IS failures. Finally, the hermeneutical perspectives of Heidegger and Gadamer play a central role in the rethinking of the relationship between technology and human thought developed by Winograd & Flores (1986).

Thus, IS research has seen applications of hermeneutic concepts in each of the levels of inquiry identified by Boland (1985). Throughout this literature, however, interpretation is approached as a solitary activity, the undertaking of a single interpreter engaged with the

¹ The hermeneutic circle is sometimes also referred to as the hermeneutic *cycle*, maintaining the concept of recursive movement between a particular and a general analysis.

relevant text. For example, in the area of systems development, it has been argued that the relevant “text” for the design effort is the flow of organizational action involving multiple actors, but the onus of interpretation in this characterization remains with a sole heroic hermeneut (Boland, 1985; Boland & Day, 1989). How then can this be reconciled with the collective pursuit of meaning-making observed in large systems development efforts? How can the hermeneutic concept inform our understanding of the ways in which a development team arrives at a common understanding of the processes they will employ? To address these questions, we explore the collective meaning-making processes that emerge in the efforts of a large systems development team. The result of this analysis is a process model of a collective hermeneutic in systems development.

Research Methodology

The setting for the current observational study is a large systems development firm that specializes in the creation of technologies to support advancements in network computing. For the purposes of the present analysis, the firm will be referred to as Amethyst Systems. Amethyst is engaged in the development of both hardware and software resources. The firm is a leading developer of network servers, personal workstations, and data storage technologies. In addition, the company has developed its own high-end line of microprocessors, which are used on many of its hardware tools. On the software side, Amethyst has developed its own operating system and a series of development platforms targeted to address the needs of developers working in an Internet environment.

The project team studied at Amethyst is responsible for the firm’s flagship platform software, Topaz. Topaz includes a full suite of tools that enable developers to create software using a programming language developed by the firm. The Topaz team has worked on several releases of the platform suite. While routine organizational turnover or movement has ensured some changes in the team’s composition, most of the development personnel have worked together on multiple releases. At the time of the data collection effort, the project team was preparing for the next release of the platform software in approximately six weeks.

Data Collection

The primary data collection method for the study was observational analysis of project team meetings. The development team is divided into sub-units based on specific task responsibilities, and the nature of the team meetings varied based on the sub-team representation and the session objectives, which included overall project discussions, “bug review” (i.e., review of requested coding changes to address perceived flaws), and engineering tasks. A total of five project meetings were observed. All meetings observed involved both collocated and geographically-distributed participants. In general, collocated team members represented three-quarters of the individuals participating in a given meeting. The average number of session participants (including distributed members) was 17. The largest session included 28 participants and the smallest involved only 10. In each of the sessions observed, multiple distinct agenda items were addressed (i.e., 3 to 4 topics per session). Table 1 provides a summary of the Topaz project team sessions observed.

Meeting Number	Duration	Primary Objective	Co-located Participants	Remote Participants	Total participants
1	2 hours	Bug review	6	4	10
2	2.5 hours	Project updates	19	2	21
3	1.25 hour	Coordination of engineering teams	5	9	14
4	2.25 hours	Project staff meeting	23	5	28
5	1.5 hours	Tech team deliverables; Bug Review	11	3	14

Table 1. Summary of project sessions observed

To capture the observational data, extensive notes were taken throughout all meetings, but no auditory or visual recordings were made. Research notes included observations regarding the specific topics discussed, styles of discourse that occurred, specific statements and the language used by meeting participants, the affective presentation of participants, the use of technical resources, and the physical layout of the meeting space. All research notes were subsequently transcribed. At the beginning of the sessions, we introduced ourselves to the meeting participants and informed them that we were conducting an observational study of the project team. We did not discuss specific research questions or the phenomena of interest.

Data Analysis

Data analysis was initiated through the coding of the transcribed observational notes. Coding was completed using NVIVO, a software package designed for the coding and analysis of qualitative research data. Following Newman & Robey (1992), we focus on individual communicative episodes as the unit of analysis. Specifically, we were concerned with the study of distinct *interpretive episodes* observed in the project team sessions. While some thinkers would argue that every social action involves interpretative processes, for the current analysis an *interpretive episode* involves a discursive event in which an explicit question of meaning or interpretation was raised. A total of 39 interpretive episodes were documented over the course of the sessions observed. The coding of interpretative episodes proceeded following an inductive process in line with a grounded theory approach (Glaser & Strauss, 1967). Episodes were analyzed with an eye to the interpretive and interpersonal mechanisms that they embodied. When novel interpretive processes were identified, an additional coding classification was created. This process was repeated until theoretical saturation was achieved (Eisenhardt, 1989; Glaser & Strauss, 1967). After the initial development of the classification system for interpretive processes, the observational notes were reviewed to determine whether any revisions were warranted to accurately represent the data.

Findings: A Model of the Collective Hermeneutic Process

The analysis of the meaning-making processes employed by the Topaz team revealed six hermeneutic mechanisms for arriving at collective understanding in the development process. These interpretive mechanisms are presented in Figure 1. While the six techniques are discussed individually, in practice they are often used in combination. Together, these techniques drive the hermeneutic circle within the team-based development environment. Employing the perspective of Paul Ricoeur (1979), we can say that the collective hermeneutic mechanisms propel the recursive dialectic between *developing interpretations* and *validating interpretations* that have been generated. In the case of the Topaz team(s), a specific interpretation is any stated understanding or definition regarding an issue discussed in the project sessions. These included interpretations with respect to the meanings of words or phrases, the objectives of a project activity, the nature of a required decision, and the understanding of processes depicted in a graphical representation. Validation of these specific interpretations entails discussion of their appropriateness in light of the *project context*, which includes the structure and overall objectives of the Topaz project, the systems development environment involved, and the broader organizational context of Amethyst.

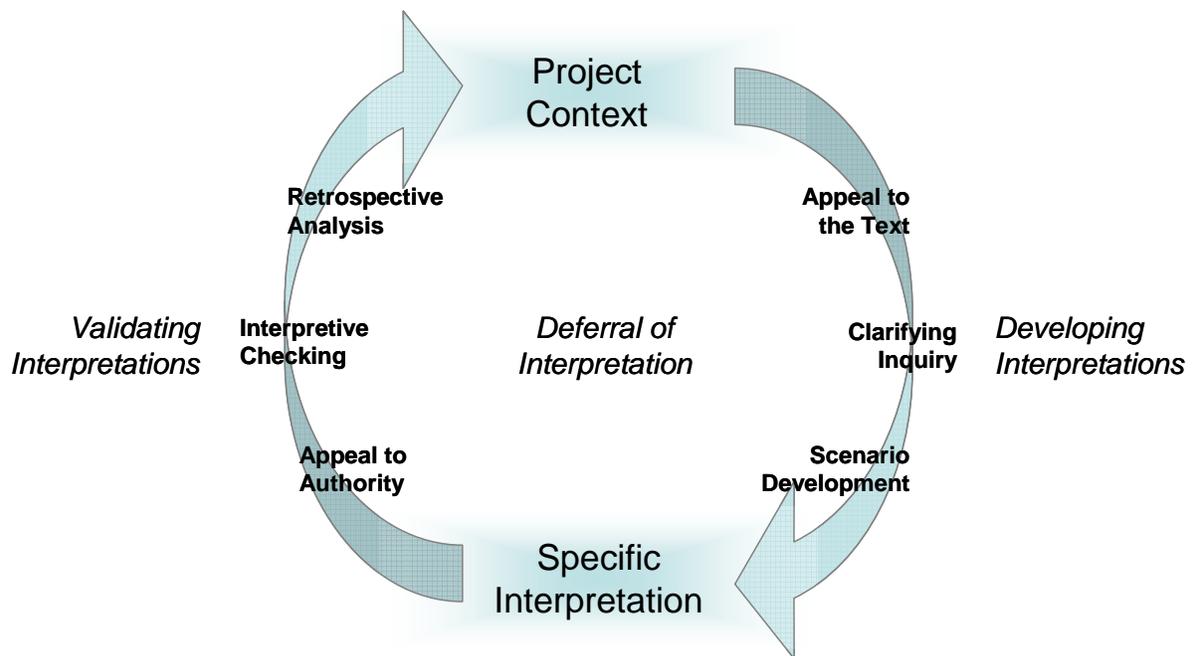


Figure 1. A model of the collective hermeneutic process

Developing Interpretations

In the interaction of the group, there were a number of processes that signaled the need for a collective interpretive effort. In several cases, these processes were initiated through an overt request for shared meaning through the fundamental question – “What do we mean by ...?” The processes employed by the groups to surface key assumptions and generate initial

interpretations of a phenomenon of interest included an *appeal to the text* supporting the project team, *clarifying inquiry* around issues that might impact an interpretation, and the *development of scenarios* to place the hermeneutical question within a practical context.

Appeal to the Text. While Ricoeur (1979) notes that systems of action are appropriate domains for hermeneutical inquiry, there is a very real sense in which the collective hermeneutic of the Topaz team was a *text-oriented* undertaking. In each of the sessions observed, external representations of the issues were provided in the form of PowerPoint presentation slides or HTML pages. Indeed, a distinct presentation file accompanied each change in topic during a project meeting. These presentation files consisted primarily of bulleted text and graphical images. Thus, in each session, much of the interpretive effort of the team was explicitly directed toward an understanding of a segment of text projected onto a screen. It is perhaps not surprising then that one of the most common interpretive techniques employed was an appeal to the text for an insight to guide interpretation. By *appeal to the text*, we are referring to occasions when one or more of the participants directs the attention of others to the external representation in response to a question about meaning. The following exchange is illustrative:

- Paul: “Is this schedule [*indicating the projected document*] reflecting the headcount you have or the headcount you’re requesting?”
 Andrea: “The headcount we’re requesting.”
 Ramesh: “Are you sure? I don’t think that’s right.”
 Paul: “Go to Slide _____. Now let’s see, [*reading aloud*] ‘Additional resources required for project.’ Now, ‘required for project’ is the part that caught my attention.”

The external representation becomes the center of the interpretive discourse. Indeed, in all of the sessions observed, the majority of the discussion was oriented toward the material projected on the screen. Furthermore, the text is understood to “speak” for a specific person or group. For example, while reviewing a bug fix request² one participant asked, “When he says ‘repeated failure,’ what exactly does he mean?” In this way, the request submitter acts as a participant in the discourse even though he or she is not actually present in the session. This phenomenon is addressed by Ricoeur (1979) when he says that writing fixes the “said” of speaking, the meaning of the speech event.

Clarifying Inquiry. Due to the task specialization that marks development work, the issue of asymmetric information regarding the specific development processes arises frequently in team discussions. On a number of occasions, meeting participants encountered content in a given presentation with which they were unfamiliar. Examples of such occurrences included questions regarding the use of a particular word or phrase to describe a task or the presentation of a process diagram which an individual had not previously encountered. When such episodes arose, the development of an explicit interpretation was often preceded by a round of *clarifying inquiry*. That is, the individual calling attention to the unfamiliar concept would ask the presenting representative and other participants for additional information before stating how he or she was understanding the issue at hand. For example:

² The subject of the “bug review” sessions, a bug fix request is a request usually submitted via email for the developers to correct or otherwise address a perceived flaw (i.e., bug) in the software.

Craig: [*Referring to an engineering process diagram*] “These nightly PITs come from the developers, or do we do that?”

Tran: “That’s from the developers.”

Craig: “So what do we do? Does your team do the weekly master updates?”

Tran: “We do the nightly PITs and overall management.”

Craig: “So it seems like there’s some redundancy here.”

While this example involves a dyad, clarifying inquiries often involved multiple project team members. The mechanism of clarifying inquiry was most pronounced in the more technically-oriented project discussions.

Scenario Development. A final prominent technique in the generation of a shared interpretation was the development of one or more realistic scenarios to help the developers “walk through” the implications of a given perspective. In the Topaz discussions, such scenario development repeatedly revolved around understanding the possible impacts on the Topaz developer community (i.e., external developers who build applications using the Topaz platform) as a result of the decisions of the project team:

Discussion of communication to developers regarding the upcoming release:

Jerry: “No one’s going to flip between the major releases in a production system. Think about it. If you were the developer, would you make that jump?”

Radha: “I can just see it. If we push these changes out without telling anyone, we will start to break people.”³

In addition, scenarios were used to explore the interpretations-in-use that may be associated with specific terms, employing real or hypothetical examples from the team members’ shared experiences to establish the context for a word or phrase:

Sheila: “What do we mean by ‘criteria’ for Beta inclusion?”

Alex: “I don’t understand what you’re asking.”

Sheila: “If someone said we have a certain ‘target,’ it would imply something we want to achieve, but that it’s not absolutely necessary. But if we say ‘criteria,’ I think most people would think that’s something that’s really required.”

Alex: “Or what happens ...?” [*in a tone implying a next logical step in the inquiry*]

Sheila: “So, these are basically guidelines then?”

Alex: “Right. That’s the way I see it.”

This focus on the natural language interpretation of terms is consistent with the work of Gadamer (1976, 2004) as well as that of the latter Wittgenstein (1953). For both of these thinkers, meaning cannot be divorced from the ways in which words or phrases are used in daily practice.

Validating Interpretation

When an interpretation has been generated, whether by an individual team member or a group, it is validated to determine the degree to which the interpretation will support appropriate action. It is important to note that validation of an interpretation is not to be equated with verification in the scientific sense. For any ‘text,’ multiple interpretations are possible. One cannot be said to be true and all others false, but validation is the process by which we assess the

³ ‘Breaking people’ refers to the introduction of a change in software that causes a significant failure for developers/users

relative fitness of a given interpretation for a particular context. Returning to Ricoeur, “it is always possible to argue for or against an interpretation, to confront interpretations, to arbitrate between them, and to seek for an agreement” (1979: 91). In the interactions of the Topaz team, there were a number of discursive mechanisms used to support this objective of validation. These included the *appeal to authority*, explicit *interpretation checking*, and *retrospective analysis*.

Appeal to Authority. Despite the collective nature of the hermeneutic process described, the reality of the workplace is that not everyone’s interpretations carry equal weight. In pursuing shared meanings for project tasks and objectives, the understandings of those with significant formal or informal authority within the group will have a disproportionately large influence on the meanings established by the group. The project manager and senior members of the project team may exercise a variety of types of power. Expert power (i.e., power derived from the unique knowledge of an individual) is of particular relevance in a systems development context (Curtis et al., 1988). Those individuals whose authority is recognized by other participants often “set” the meaning that is to be shared by all group members.

In the Topaz sessions, the power structures of the group were apparent in the way in which the meaning-making efforts were directed. Questions over the validity of interpretations were generally directed to the project manager or the manager of one of the engineering teams:

Mike: [*To the project manager*] “How should we tag⁴ various kinds of bugs? I’m assuming we can use the labels we’ve been using.”

Craig: “No, you want to be using the tags that I posted in the EC charts. Has everyone seen the EC charts that I created?” [*other participants acknowledge receipt*] “So, use the tags I have there.”

Mike: “So, should we go back to retag some of the older requests?”

Craig: “Well, no. But going forward, we want to use these tags.”

Mike: “Understood.”

In addition, managers are direct in advising interpretations and behaviors that would be appropriate in a specific context:

Ralph:[*To the engineering team lead*] “We’re not using the same format for our process as Monica is. [*Referring to different process diagrams employed by engineering sub-teams*]. Is that alright?”

Alice: [*Engineering team lead*]: “For these meetings, I think we need to have a more consistent presentation format, so everyone can understand what everyone else is doing. I like the workflow diagram, so if we could all start using that diagram, that would be great. That’s the format Anjun was using.”

Interpretation Checking. Just as the generation of an interpretation was frequently initiated with an overt question regarding meaning, the validation of an interpretation can be undertaken through a request for input from the group on an interpretation that has been generated. *Interpretation checking* refers to a discursive event in which a participant in the meeting outlines his or her understanding of a issue and asks the other participants for feedback on the stated interpretation. Such checking efforts were observed repeatedly throughout the sessions. In general, they were marked by a simple question posed to all participants:

⁴ Tagging refers to the way in which the development team classifies various types of bugs. The tags used for bugs impact the perceived urgency of bug fix efforts.

Jin: "So there are no significant issues with the feature audit. Is that accurate?" [No response from other participants] "OK."

Sue: "Am I misunderstanding something? M_____ is just a query tool, isn't it?"

Through this simple process of checking, the development team is able to assess the degree to which they have a shared understanding of the topics they have discussed. In addition, checking provides an opportunity for revising one's individual interpretations in light of the perspectives offered by other team members. To employ the language favored by Gadamer, soliciting the group's input presents a mechanism for identifying one's own prejudices and incorporating that recognition into subsequent interpretive efforts.

Retrospective Analysis. Reminiscent of the sense-making perspective of Weick (1979), members of the Topaz team at times looked to past action to validate contemporary interpretive efforts. *Retrospective analysis* reflects a situation in which an interpretive effort is informed by earlier interpretations encoded in organizational actions or policies. When members of a development team look to the ways in which specific processes have been pursued in earlier projects as a guide to future action, they are supporting their interpretation through retrospective analysis:

Blake: "What do we mean by 'repeatable crash'?"

Joel: "I don't know ... 'repeatable crash.'" (*Laughing*)

Blake: "Well, what do we mean by 'crash'? See, I know of a crash in Topaz that has not been resolved to my knowledge. So I'm saying that to me that's a crash ... a *repeatable* crash."

Joel: "But it's not a showstopper, because we didn't stop the show when we decided to go forward with the release."

Blake: "So, that's not what we have in mind?"

Joel: "I would say no."

However, in contrast to Weick's vision, while the retrospective analysis is building upon past experience, it is oriented toward making sense of future action. The interpretations developed by the project team are intended to establish a base of shared understanding for coordination in subsequent development efforts.

Deferral of Interpretation

In the analysis of the Topaz team, a final discursive mechanism was identified which does not belong within the hermeneutic process of the group. At the same time, it might rightly be called a part of the interpretive agenda of the development effort. In this study, we shall refer to this phenomenon as the deferral of interpretation. Such a deferral is represented by episodes in which the project team acknowledges the need for shared meaning around some facet of the development process, but chooses not to pursue the achievement of common understanding at the time the issue is raised. In situations of deferred interpretation, the team members identify significant impediments to the resolution of contested understandings, but they do not perceive the absence of shared meaning as an encumbrance to near-term action. This dynamic is illustrated in the following two examples:

- Simon: “When we can’t come up with a definition for how we translate our bug states into messages, I’m not sure I’m comfortable exposing those things to customers. I mean do we want customers to see some of these things?”
- Helen: “I will give you that point. Four hundred of us can’t agree about what a bug state means. So maybe we don’t want to expose that inconsistency. But maybe that’s the way we’re going to get it fixed.”
- Alex: “The problem is we don’t have a definition of what ‘priority’ means.”
- Mike: “Well, we do, but there are differences of opinion on it. N engineers may have n times 2 opinions about what ‘priority’ means.” [*Discussion moves to the next topic*]

On the Topaz team, when an interpretation was deferred, no specific schedule was set for returning to the issue. By combining the acknowledgement of discrepancy with the decision not to seek resolution, the deferral of interpretation can be understood as a “tabling” of contested issues for discussion at some undetermined later date.

“Roles” and the Hermeneutic Techniques

The interpretive mechanisms outlined here represent a repertoire of hermeneutic techniques employed by the Topaz team. In developing the collective hermeneutic model, individual techniques have been presented in association with the “role” they played in the interpretive efforts (i.e., generating or validating interpretations). However, we do not argue that any one of these methods is strictly isolated to the generation or validation phases of a hermeneutic circle. For example, while an *appeal to the text* was most commonly employed in the generation of novel interpretations, it is conceivable that this same method could be used to arbitrate between competing interpretations, thereby functioning as a mechanism for validation. In large part, this reflects the fact that the generating and validating facets of the hermeneutic process are not as easily distinguished in practice as they are in theory. To adopt a phrase from Cook & Brown (1999), the movement of the hermeneutic circle might be understood as a *generative dance* between the understanding of the particular and the understanding of the general – between guessing and validating. At any given time, it may be difficult to determine which is leading.

Discussion

One of the striking aspects of the Topaz study is the amount of time and effort that was directed to explicitly hermeneutical activity. In just five project teams meetings, we encountered nearly 40 episodes of explicit discussion or debate around the language and images used to drive the conversation. In many of these interpretive episodes, the discussions were initiated through questions concerning basic concepts or language:

- “What do we mean by ‘criteria’?”
- “Now, when do we want to say ‘compatible’ as opposed to ‘conformant’?”
- “How do we tag various kinds of bugs?”
- “What do we mean by ‘repeatable crash’?”
- “Wait, wait. Who is ‘we’?”
- “Do typos in GUI count as an ‘embarrassing’ bug?”
- “What’s the difference between ‘severe’ and ‘significant’?”

The fundamental nature of these debates is particularly intriguing in light of the fact that the Topaz team was not a newly-formed project team but had worked together through multiple releases of the platform. This stands in contrast to much of systems development work, in which individual project teams are pulled together on an ad hoc basis to ensure an appropriate combination of technical skills (Kraut & Streeter, 1995; Tellioglu & Wagner, 1999). If we accept the contention that repeated interaction over time leads to the development of shared mental models and corresponding improvements in performance (Crowston & Kammerer, 1998; Weick & Roberts, 1993), the interpretive demands of groups with a less extensive history must be much greater than those encountered by the Topaz team.

Interestingly, the observations with respect to the interpretive workload of the Topaz group are consistent with the hermeneutic concepts explored by Gadamer (1976; Gadamer, 2004). For Gadamer, the demand for hermeneutic analysis is universal and continuous. The process of interpretation is repeated because the prejudices that individuals bring to the text are not static but can change over time. Thus, the meaning that the interpreter takes from his or her conversation with the text can change as well:

“The real meaning of a text, as it speaks to the interpreter, does not depend on the contingencies of the author and his original audience. It certainly is not identical with them, for it is always co-determined also by the historical situation of the interpreter ... Not just occasionally but always, the meaning of a text goes beyond its author. That is why understanding is not merely a reproductive but always a productive activity as well ... It is enough to say that we understand in a *different way, if we understand at all*” (Gadamer, 2004: 296).

Adopting this perspective, it should not be surprising that a well-established development team continues to engage in the process of interpretation around a variety of concepts. As their collective experiences change, the prejudices that they bring to the development process will change as well. The differences in understanding that emerge may not parallel those that compete across generations of textual analysis, but they are significant enough to engender repeated exploration of some seemingly basic ideas.

The interactions of the Topaz team suggest that the exercise of a collective hermeneutic process is one of the primary objectives of the meeting structure itself. In more than half of the agenda discussions (i.e., 10 of the 18 distinct presentation topics) conducted in the observed meetings, the materials used in the presentation had been distributed to all session participants in advance. Given this dissemination of information, the team members ostensibly had equal basis for establishing an individual interpretation. Yet by coming together in a meeting, the project team is able to engage in an interpretive process collectively. This ensures that, to the degree possible, the members of the project team truly *share* the interpretations that are developed.

This collective hermeneutic perspective presents several implications for the support of an interpretive systems development process. In general, it suggests that the productivity and effectiveness of systems development teams might be enhanced through additional avenues for collaborative exploration of meaning by the group. For example, in a widely cited article, Boland and Tenkasi (1995) present a vision of the ways in which information systems can be designed to support the goals of *perspective making* and *perspective taking* within knowledge communities. Perspective making refers to the development of a common base of knowledge within a given community, and perspective taking is the exchange of understandings between distinct communities. Boland & Tenkasi (1995) focus on the potential role of boundary objects (Star, 1990) in supporting such knowledge formation and communication. Specific measures

they suggest include the development of groupware forums for the articulation of task narratives, the exchange of representations (e.g., text documents, process diagrams), and the reflective expression of interpretive readings. In the domain of systems development teams, resources of this nature could be beneficial in driving collective hermeneutic behavior outside the context of formal meetings.

It is important to note that development groups embody both the perspective making and perspective taking aspects of the vision developed by Boland & Tenkasi (1995). As we have discussed, project teams represent a single community around the common initiative, but they are comprised of representatives from multiple technical and functional knowledge communities. In addition, it is critical to distinguish that the collective hermeneutic process outlined in this study focuses on the continual development of new interpretations *within* any given knowledge community, rather than assuming a static interpretive scheme that has been formed and solidified by that community. Therefore, the forum resources proposed would have to support an ongoing conversation *between* members of a community as well as across community boundaries.

Other developments from the groupware and computer supported cooperative work (CSCW) traditions could enhance the exercise of a collective hermeneutic process as well. The development of collaborative virtual workspaces (Maybury, 2001; Spellman, Mosier, Deus, & Carlson, 1997) could create a computer-mediated context for the interpretive conversations required in the hermeneutic process. Such environments would be especially useful for supporting the interpretive interaction of project teams with significant geographic distribution. Virtual workspaces would have to be designed with an emphasis on reinforcing both the *communicative* and *discursive* forms of action observed in group collaboration (Ngwenyama & Lyytinen, 1997).⁵

Finally, simple web-based collaborative methods such as a wiki environment (Leuf & Cunningham, 2001) hold significant promise for supporting the type of collaborative process outlined in this study. First developed in the early-1990s, *wiki* is an open source server software that allows all users to edit web page content using any type of web browser. Additional important features of wiki include the ability to rapidly develop new pages and crosslink them to existing sources within the domain and the maintenance of an archive of version changes that a single document/page has undergone since its inception. While the setup of such an environment is extremely simple, the implications for collective hermeneutic behavior are significant. As noted above, several of the presentation files used in the Topaz project team meetings were in an HTML format and discussions arose around terms or concepts that appeared on these pages. If such meetings were enabled with a wiki environment, the project team could immediately update the documents based on the interpretive outcomes of the session, including the development of a new linked page to capture the discussion that took place around a given concept.

To focus on a specific example, if discussion over the meaning of the word ‘criteria’ were to arise, the project team could document key points of the conversation and the resolution that they achieved through the spontaneous development of a new page linked to the term ‘criteria’ in the primary project document(s). In subsequent team discussions, if the question of ‘criteria’ were to resurface, the new resource could be used as point of reference for the interpretive discourse. It would not however inhibit the development of a new – or revised – interpretation

⁵ Ngwenyama & Lyytinen (1997) build upon the categories of social action developed in the critical theory of Jurgen Habermas. In this context, *communicative actions* refer to those aimed at maintaining mutual understandings and coordinating action within a group, whereas *discursive action* involves critical debate and is directed at re-establishing consensus.

based on changes in the experiences, sensitivities, or composition of the development team. Through repeated development and alteration of such resources, the project team would build up a collective memory. Indeed, the development of improved tools to support team memory has been cited as a crucial need in the management of systems development efforts (Walz et al., 1993).

This type of project resource need not be limited to the context of project meetings. For a given wiki, the right to edit pages can be limited to select members of an organization (e.g., project team members). Since relevant members of the team could update the project documents at all times, the project's wiki environment could become a locus for the exchange of individual interpretations and the development of shared understandings throughout the life of the project. Furthermore, it could help to identify areas where interpretive discussions needed to occur because of the presence of disparate understandings of the problem space. Such a practice could achieve many of the ends of the interpretive reading forums proposed by Boland & Tenkasi (1995) without the demand for overt reflexivity that such forums imply.

In addition to providing insights for the enhancement of contemporary systems development environments, the collective hermeneutic process modeled in this study holds significant implications for research on information systems design and use. If we acknowledge the inherently social nature of systems development, then we must recognize that interpretation itself occurs through social mechanisms. The collective hermeneutic model extends the hermeneutic tradition in IS research by addressing the ways in which interpretation takes shape not simply within the mind of an individual but also in collaboration with other interpreters. In addition, the model provides a perspective for breaking open the black box of shared understanding that emerges within a heterogeneous community.

Limitations

This study is an initial attempt at exploring the process of collective interpretation within a systems development environment. As an exploratory research effort, the study has a number of limitations that must be acknowledged. First, the study reflects hermeneutic analysis at two distinct levels – the interpretive activities of a systems development team and the researchers' own interpretations of that hermeneutic process. As a result of this double hermeneutic, the study presents certain impediments to the more systematic methods of assessment and verification. To borrow an idea from Clifford Geertz (1973), “right down to the factual case, the hard rock, insofar as there is any, of the whole enterprise, we are already explicating; and worse, explicating explications” (p. 9). Thus, we do not make the claim in this study that the collective hermeneutic process outlined here is generalizable to all other systems development contexts, much less to project environments more broadly defined. However, we do claim that the collective hermeneutic model provides an important foundation for further exploration of the interpretive forces at play on systems development teams.

The potential for additional research points to the other central limitation of the current study. With only five primary data collection sessions and 39 interpretive episodes captured, the relatively small sample size limits the degree to which the process observed can be claimed to be universal within the research setting. The observations were focused on the workings of one large development team (albeit, incorporating multiple sub-teams) within a single organizational context. The ideal of a thorough comparative analysis in the grounded theory approach (Glaser & Strauss, 1967) suggests that the study's findings could be significantly strengthened if similar

patterns of group discourse and interpretive behavior were observed in other development environments. In particular, additional research efforts could focus on observation of teams in an in-house development setting (as opposed to an IT vendor environment like that of the Topaz group). In-house development efforts occur in nearly every industrial sector, and a theoretical sampling from across this spectrum would enable the assessment of possible changes to the interpretive process based on the differences in the business contexts of systems development units.

Conclusion

The inherent complexity of large systems development efforts creates significant impediments to the establishment of shared meaning between members of a project team. Yet communication and coordination of effort remain essential for the effective execution of systems development initiatives. In this study, we have developed a model of the meaning-making efforts of a project team that is based on the concept of a collective hermeneutic process. Through its concerted application of a shared repertoire of interpretive techniques, the Topaz team reveals the ways in which a development team can work together toward resolving the ambiguities of their day-to-day collaboration.

The collective hermeneutic model provides a foundational framework for further research into the meaning-making processes of development groups. As an exploratory study, this research calls for assessment of the collective hermeneutic model through application to a wider variety of systems development contexts. Are the interpretive processes observed in the creation of a large platform system applicable to the efforts of smaller project teams or those of in-house development projects? How does the degree of geographic distribution of a project team impact the hermeneutic techniques that its members employ? Similarly, what effect does the business context of a firm have on the interpretive mechanisms of its systems development personnel? The investigation of such issues will offer us a better understanding of how IS professionals come together around a shared vision of the future.

References

- Boehm, B. W. 1988. A Spiral Model of Software Development and Enhancement. Computer, 21(5): 61-72.
- Boland, J., Richard J. 1985. Phenomenology: A Preferred Approach to Research on Information Systems. In E. Mumford & R. Hirschheim & G. Fitzgerald & A. T. Woods-Harper (Eds.), Research Methods in Information Systems: 193-201. Amsterdam: North-Holland.
- Boland, J., Richard J. 1987. The Coming Hermeneutic Turn in Accounting Research. Paper presented at the 8th Annual Big Ten Accounting Doctoral Consortium, University of Illinois.
- Boland, J., Richard J. & Day, w., F. 1989. The Experience of System Design: A Hermeneutic of Organizational Action. Scandinavian Journal of Management, 5(2): 87-104.

- Boland, J., Richard J. 1991. Information System Use as a Hermeneutic Process. In H.-E. Nissen & H. K. Klein & R. Hirschheim (Eds.), Information Systems Research: Contemporary Approaches and Emergent Traditions. Amsterdam: North-Holland.
- Boland, J., Richard J. & Tenkasi, R. V. 1995. Perspective Making and Perspective Taking in Communities of Knowing. Organization Science, 6(4): 350-372.
- Brooks, J., Frederick P. 1987. No Silver Bullet: Essence and Accidents of Software Engineering. Computer, 20(4): 10-19.
- Carmel, E. 1997. American Hegemony in Packaged Software Trade and the "Culture of Software". The Information Society, 13: 125-142.
- Chen, A. Y. S., Sawyers, R. B., & Williams, P. F. 1997. Reinforcing Ethical Decision Making Through Corporate Culture. Journal of Business Ethics, 16(8): 855-865.
- Cook, S. D. N. & Brown, J. S. 1999. Bridging Epistemologies: The Generative Dance Between Organizational Knowledge and Organizational Knowing. Organization Science, 10(4): 381-400.
- Crowston, K. & Kammerer, E. E. 1998. Coordination and Collective Mind in Software Requirements Development. IBM Systems Journal, 37(2): 227-245.
- Curtis, B., Krasner, H., & Iscoe, N. 1988. A Field Study of the Software Design Process for Large Systems. Communications of the ACM, 31(11): 1268-1287.
- Daft, R. L. & Weick, K. E. 1984. Toward a Model of Organizations as Interpretive Systems. Academy of Management Review, 9(2): 284-295.
- Davis, G. B., Lee, A. S., Nickles, K. R., Chatterjee, S., Hartung, R., & Wu, Y. 1992. Diagnosis of an Information System Failure. Information & Management, 23: 293-318.
- Donnellon, A., Gray, B., & Bougon, M. G. 1986. Communication, Meaning, and Organized Action. Administrative Science Quarterly, 31(1): 43-55.
- Eisenhardt, K. M. 1989. Building Theories from Case Study Research. Academy of Management Review, 14(4): 532-550.
- Ewusi-Mensah, K. 1997. Critical Issues in Abandoned Information Systems Development Projects. Communications of the ACM, 40(9): 74-80.
- Faraj, S. & Sproull, L. 2000. Coordinating Expertise in Software Development Teams. Management Science, 46(12): 1554-1568.
- French, J. R. P., Jr. & Raven, B. 1959. The Bases of Social Power. In D. Cartwright (Ed.), Studies in Social Power. Ann Arbor, MI: Institute for Social Research, University of Michigan.
- Gadamer, H.-G. 1976. Philosophical Hermeneutics. Berkeley, CA: University of California Press.
- Gadamer, H.-G. 2004. Truth and Method (J. Weinsheimer & D. G. Marshall, Trans.) (2ne, Revised ed.). London: Continuum.
- Geertz, C. 1973. The Interpretation of Cultures. New York: Basic Books, Inc.

- Glaser, B. G. & Strauss, A. L. 1967. The Discovery of Grounded Theory: Strategies for Qualitative Research. Chicago, IL: Aldine Publishing Company.
- Goodenough, W. 1964. Cultural Anthropology and Linguistics. In D. H. Hymes (Ed.), Language in Culture and Society: A Reader in Linguistics and Anthropology. New York: Harper & Row.
- Grondin, J. 1991. Introduction to Philosophical Hermeneutics. New Haven, CT: Yale University Press.
- Guinan, P. J., Coopriker, J. G., & Faraj, S. 1998. Enabling Software Development Team Performance During Requirements Definition: A Behavioral Versus Technical Approach. Information Systems Research, 9(2): 101-125.
- Heidegger, M. 1962. Being and Time (J. MacQuarrie & E. Robinson, Trans.) (1st English Edition ed.). London: SCM Press.
- Hirschheim, R. & Klein, H. K. 1989. Four Paradigms of Information Systems Development. Communications of the ACM, 32(10): 1199-1216.
- Hirschheim, R., Klein, H. K., & Lyytinen, K. 1995. Information Systems Development and Data Modeling. Cambridge, UK: Cambridge University Press.
- Katz, D. & Kahn, R. L. 1978. The Social Psychology of Organizations (2nd ed.). New York: Wiley.
- Kraut, R. F. & Streeter, L. A. 1995. Coordination in Software development. Communications of the ACM, 38(3): 69-81.
- Lee, A. S. 1991. Integrating Positivist and Interpretive Approaches to Organizational Research. Organization Science, 2(4): 342-365.
- Leuf, B. & Cunningham, W. 2001. The Wiki Way: Quick Collaboration on the Web. Boston: Addison-Wesley.
- Levesque, L. L., Wilson, J. M., & Wholey, D. R. 2001. Cognitive Divergence and Shared Mental Models in Software Development Project teams. Journal of Organizational Behavior, 22: 135-144.
- Levina, N. & Vaast, E. 2005. The Emergence of Boundary Spanning Competence in Practice: Implications for Implementation and Use of Information Systems. MIS Quarterly, 29(2): 335-363.
- Maines, D. R. 2000. The Social Construction of Meaning. Contemporary Sociology, 29(4): 577-584.
- Maybury, M. 2001. Collaborative Virtual Environments for Analysis and Decision Support. Communications of the ACM, 44(12): 51-54.
- Myers, M. D. 1994. A Disaster for Everyone to See: An Interpretive Analysis of a Failed IS project. Accounting, Management, and Information Technology, 4(4): 185-201.
- Myers, M. D.; Qualitative Research in Information Systems; <http://www.qual.auckland.ac.nz/>.
- Newman, M. & Robey, D. 1992. A Social Process Model of User-Analyst Relationships. MIS Quarterly, 16(2): 249-266.

- Ngwenyama, O. K. & Lyytinen, K. J. 1997. Groupware Environments as Action Constitutive Resources: A Social Action Framework for Analyzing Groupware Technologies. Computer Supported Cooperative Work, 6(1): 71-93.
- Ricoeur, P. 1979. The Model of the Text: Meaningful Action Considered as a text. In P. Rabinow & W. M. Sullivan (Eds.), Interpretive Social Science: A Reader. Berkeley, CA: University of California Press.
- Sawyer, S., Farber, J., & Spillers, R. 1996. Supporting the Social Processes of Software Development. Information Technology & People, 10(1): 46-62.
- Sawyer, S. & Guinan, P. J. 1998. Software Development: Processes and Performance. IBM Systems Journal, 37(4): 552-569.
- Smircich, L. 1983. Concepts of Culture and Organizational Analysis. Administrative Science Quarterly, 28(3): 339-358.
- Spellman, P. J., Mosier, J. N., Deus, L. M., & Carlson, J. A. 1997. Collaborative Virtual Workspace. Paper presented at the International ACM SIGGROUP Conference of Supporting Group Work., Phoenix, AZ.
- Star, S. L. 1990. The structure of ill-structured solutions: boundary objects and heterogeneous distributed problem solving, Distributed Artificial Intelligence, Vol. 2: 37-54. San Francisco, CA: Morgan Kaufmann Publishers.
- Teasley, S. D., Covi, L. A., Krishnan, M. A., & Olson, J. S. 2002. Rapid Software Development through Team Collocation. IEEE Transactions on Software Engineering, 28(7): 671-683.
- Tellioglu, H. & Wagner, I. 1999. Software Cultures: Exploring Cultural Practices in Managing Heterogeneity within System Design. Communications of the ACM, 42(12): 71-77.
- Tice, T. N. & Slavens, T. P. 1983. Research Guide to Philosophy. Chicago: American Library Association.
- Tichy, N. M. 1982. Managing Change Strategically: The Technical, Political, and Cultural Keys. Organizational Dynamics, 11(2): 59-80.
- Walz, D. B., Elam, J. J., & Curtis, B. 1993. Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration. Communications of the ACM, 36(10): 63-77.
- Weick, K. E. 1979. The Social Psychology of Organizing. New York: Random House.
- Weick, K. E. & Roberts, K. H. 1993. Collective Mind in Organizations: Heedful Interrelating on Flight Decks. Administrative Science Quarterly, 38(3): 357-381.
- Winograd, T. & Flores, F. 1986. Understanding Computers and Cognition: A New Foundation for Design. Norwood, NJ: Ablex Publishing Company.
- Wittgenstein, L. 1953. Philosophical Investigations (G. E. M. Anscombe, Trans.). New York: Macmillan.

Editors:

Michel Avital, University of Amsterdam
Kevin Crowston, Syracuse University

Advisory Board:

Kalle Lyytinen, Case Western Reserve University
Roger Clarke, Australian National University
Sue Conger, University of Dallas
Marco De Marco, Università Cattolica di Milano
Guy Fitzgerald, Brunel University
Rudy Hirschheim, Louisiana State University
Blake Ives, University of Houston
Sirkka Jarvenpaa, University of Texas at Austin
John King, University of Michigan
Rik Maes, University of Amsterdam
Dan Robey, Georgia State University
Frantz Rowe, University of Nantes
Detmar Straub, Georgia State University
Richard T. Watson, University of Georgia
Ron Weber, Monash University
Kwok Kee Wei, City University of Hong Kong

Sponsors:

Association for Information Systems (AIS)
AIM
itAIS
Addis Ababa University, Ethiopia
American University, USA
Case Western Reserve University, USA
City University of Hong Kong, China
Copenhagen Business School, Denmark
Hanken School of Economics, Finland
Helsinki School of Economics, Finland
Indiana University, USA
Katholieke Universiteit Leuven, Belgium
Lancaster University, UK
Leeds Metropolitan University, UK
National University of Ireland Galway, Ireland
New York University, USA
Pennsylvania State University, USA
Pepperdine University, USA
Syracuse University, USA
University of Amsterdam, Netherlands
University of Dallas, USA
University of Georgia, USA
University of Groningen, Netherlands
University of Limerick, Ireland
University of Oslo, Norway
University of San Francisco, USA
University of Washington, USA
Victoria University of Wellington, New Zealand
Viktoria Institute, Sweden

Editorial Board:

Margunn Aanestad, University of Oslo
Steven Alter, University of San Francisco
Egon Berghout, University of Groningen
Bo-Christer Bjork, Hanken School of Economics
Tony Bryant, Leeds Metropolitan University
Erran Carmel, American University
Kieran Conboy, National U. of Ireland Galway
Jan Damsgaard, Copenhagen Business School
Robert Davison, City University of Hong Kong
Guido Dedene, Katholieke Universiteit Leuven
Alan Dennis, Indiana University
Brian Fitzgerald, University of Limerick
Ole Hanseth, University of Oslo
Ola Henfridsson, Viktoria Institute
Sid Huff, Victoria University of Wellington
Ard Huizing, University of Amsterdam
Lucas Introna, Lancaster University
Panos Ipeirotis, New York University
Robert Mason, University of Washington
John Mooney, Pepperdine University
Steve Sawyer, Pennsylvania State University
Virpi Tuunainen, Helsinki School of Economics
Francesco Virili, Università degli Studi di Cassino

Managing Editor:

Bas Smit, University of Amsterdam

Office:

Sprouts
University of Amsterdam
Roetersstraat 11, Room E 2.74
1018 WB Amsterdam, Netherlands
Email: admin@sprouts.aisnet.org