Journal of Information Technology (2013) 28, 198–223

palgrave-journals.com/jit/

Research article

Language quality in requirements development: tracing communication in the process of information systems development

Christoph Rosenkranz, Marianne Corvera Charaf, Roland Holten

Goethe University, Frankfurt am Main, Germany

Correspondence:
C Rosenkranz, Goethe University, Faculty of Business Administration and Economics, RuW Building, Grüneburgplatz 1, Frankfurt am Main D-60325, Germany.
Tel: +49 (0) 69-798-34674;
Fax: +49 (0) 69-798-35005;
E-mail: rosenkranz@wiwi.uni-frankfurt.de

Abstract

Knowledge transfer, communication, and shared understanding between project stakeholders are important factors in requirements development and in the information systems development process. Nevertheless, the impact and analysis of language and linguistic communication during requirements development is still an open issue. In our research, we claim that requirements development depends on the ability to deal with language and communication issues in practice and reach shared understanding of requirements. We propose the concept of language quality as a suitable means for analyzing the emergence of coherent and meaningful requirements. By applying the thereby developed dimensions of language quality to a real information systems development project, we are able to obtain practice-grounded propositions to further evaluate the consequences of different actions on the interaction and communication process of stakeholders in requirements development.

Journal of Information Technology (2013) **28,** 198–223. doi:10.1057/jit.2012.33; published online 15 January 2013

Keywords: information systems development process; requirements development; requirements engineering; language quality; communication quality; shared understanding

Introduction

A large body of knowledge exists on different aspects of *information systems development* (ISD) (Hirschheim *et al*., 1995; Iivari *et al*., 2004). Academics and practitioners have proposed many methodologies and practices for specifying or designing software and software-intensive information systems (for example, Boehm, 1988; Davis, 1990; Fichman and Kemerer, 1993; Loucopoulos and Karakostas, 1995; Morrison and George, 1995; Galliers and Swan, 2000; Sommerville, 2001; Beck and Andres, 2004). Software and IT-based information systems (IS) are often developed in projects using these approaches (Hirschheim *et al*., 1995: 33). Many difficulties in these projects arise from the intangible nature of software (Cule *et al*., 2000: 65) that leads to difficulties in coordinating activities during the ISD process (Kraut and Streeter, 1995), to uncertainties and ambiguities influencing the result, to increased costs and, in some cases, to failure of the whole process (Land and Somogyi, 1986; Kraut and Streeter, 1995; Cule *et al*., 2000; Xia and Lee, 2005). IS researchers have investigated these difficulties over many years by adapting different perspectives (for example, Bostrom and Heinen, 1977; Guinan and Bostrom, 1986; Orlikowski and Robey, 1991; Newman and Robey, 1992; Hirschheim *et al*., 1995; Sawyer and Guinan, 1998; Lyytinen and Robey, 1999; Watts Sussman and Guinan, 1999; Sambamurthy and Kirsch, 2000; Gallivan and Keil, 2003; Kautz and Nielsen, 2004; Chae and Poole, 2005; Ko *et al*., 2005; Vidgen and Wang, 2009). The need for research remains, however, as various studies still point to recurring problems arising in ISD, such as cost or time overruns, rollouts with fewer features than promised, or total failures (Avison and Fitzgerald, 2003; Agrawal and Chari, 2007; Nelson, 2007). Despite different circumstances, the basic message has persisted over the past five decades: the challenge of developing reliable software and software-intensive IS in a controlled and cost-effective way still exists (Boehm and Basili, 2000; Berry, 2004; Kautz *et al*., 2007; Fraser and Mancl, 2008; Jarke *et al*., 2009).

The major problems of ISD are 'not so much technological as sociological in nature' (DeMarco and Lister, 1987: 4), not least because IS at the core are 'social systems that are technically implemented' (Hirschheim *et al*., 1995: 1). ISD is characterized by multiple stakeholders and multiple influences that are related to pre-existing IT systems and to organizational, cultural, and social elements that have evolved in organizations over decades (Chae and Poole, 2005). Fundamentally, ISD is a *social process* that is carried out in an organizational setting and that involves users, systems analysts, developers, and other stakeholders such as managers or customers, as well as their social dynamics (Newman and Robey, 1992; Hirschheim *et al*., 1995; Robey and Newman, 1996; Truex *et al*., 1999; Lyytinen and Newman, 2008). Major drivers for ISD success (Siau *et al*., 2010) are knowledge transfer (Robillard, 1999; Ko *et al*., 2005; Levina and Vaast, 2005; Joshi *et al*., 2007; Williams, 2010), communication (Guinan and Bostrom, 1986; Bostrom, 1989; Gallivan and Keil, 2003), and shared understanding (Tan, 1994; He *et al*., 2007), all of which are related to the social interactions between stakeholders.

Specifically, the communication, negotiation, and determination of requirements during the phase of requirements development have a central role (Davidson, 2002; Ambriola and Gervasi, 2006; Mathiassen *et al*., 2007; Chakraborty *et al*., 2010). *Requirements development* deals with the elicitation, analysis, specification, documentation, and validation of stakeholder requirements that are to be met by software-intensive IT systems (Abran *et al*., 2004). It is a process that tries to create coherent and meaningful specifications of software and systems by the consolidation of different stakeholders' perspectives and multiple requirements in an organization (Curtis *et al*., 1988). These specifications are a definition of the envisioned result. The subsequent ISD phases try to design and implement an IT system that meets this vision. Consequently, it is crucial for requirements development to understand how different stakeholders reach consensus on coherent and meaningful requirements, and to track how they create a shared understanding of the requirements (Hansen and Rennecker, 2006).

We argue that it is necessary to investigate the *social interactions of stakeholders* to describe, explain, and understand this process. However, there is a general paucity of theory and studies of social behavior and processes in ISD (Sambamurthy and Kirsch, 2000; Kautz *et al*., 2007). The ISD process 'has received relatively little research attention in prior literature' (Siau *et al*., 2010: 88), and only a few researchers have examined social interactions in detail (for example, Guinan and Scudder, 1989; Newman and Robey, 1992; Sawyer and Guinan, 1998; Watts Sussman and Guinan, 1999; Sawyer *et al*., 2010). Our primary goal with this article is to contribute a novel perspective that begins to open the 'black box' (Siau *et al*., 2010: 92) of the ISD process, and to examine the building of consensus and shared understanding of requirements through social interactions. To this end, we build on our previous research (Corvera Charaf *et al*., 2012) and study social interactions during requirements development as a *language development process* (Lyytinen, 1985: 61). Social interactions are connected to human communication primarily through the medium of language (Hirschheim *et al*., 1995: 1), and social interactions become visible when linguistic communication is observed: social encounters are sustained above all through talk, through everyday conversation using language (Giddens, 1984: 73, 264). Furthermore, communication involves more than just words or verbal, written or spoken, conversations. Communicative acts such as vocal gestures, facial expressions, eye gaze, and postures also help people to make signals and exchange messages in face-to-face situations (Clark, 1996: 188; Tomasello, 2008: 274).

The remainder of this article is structured as follows. First, we analyze related work in the field of ISD and requirements development. We suggest the concept of language quality for analyzing the language development process and the emergence of coherent and meaningful requirements. Afterwards, we outline our research methodology. We gathered data of linguistic communication between stakeholders in an observational field study, where we observed the phase of requirements development in an ISD project. By applying the concept of language quality to authentic communication data obtained from the field study, we are able to induce propositions about the language development process. We obtain practice-grounded propositions for practitioners to consider and for researchers to further evaluate the consequences of different actions on social interactions and communication during ISD. After discussing the implications of our findings, we finish the article by outlining further research.

Related work and theoretical background

Communication in ISD and requirements development

Methodologies for developing software and software-intensive IS range from sequential (waterfall model, Royce, 1970) to iterative (spiral model, Boehm, 1988), experimental (prototyping, Maryam, 1984), and agile approaches (Extreme Programming, Beck and Andres, 2004; Lean, Poppendieck and Poppendieck, 2003; or Scrum, Schwaber, 1995). Most sequential or iterative methodologies are plan driven and rely on *formal communication*, such as models and documents, among participants (Kraut and Streeter, 1995; Boehm and Turner, 2004; Black *et al*., 2009). For example, requirements are usually stated within a requirements document, which at the end of the analysis phase is a specification of the IT system to be built (Pohl, 1994). In rapidly changing environments, however, formal mechanisms of communication based on models or documents are not flexible enough to react quickly to changes (Byrd *et al*., 1992; Kraut and Streeter, 1995; Herbsleb and Mockus, 2003): 'Rather than being bastions of order in an uncertain world, traditional teams may indeed become chaotic should their plan-driven organization be overwhelmed by events' (Vidgen and Wang, 2009: 374). By contrast, many agile methodologies suggest that human and social factors are important (Dybå and Dingsøyr, 2009); business customers and developers should work together daily, and project information should be shared through *informal communication* such as frequent face-to-face conversation rather than through documentation and models. Some agile approaches rely more or less entirely on oral communication (Ramesh *et al*., 2010: 451).

Regardless of the use of either traditional or agile approaches, the *requirements development process* shows many problems and is still an important field of research (Jarke *et al*., 2009). The environment now involves elements that were not there 20 or 30 years ago (Hansen and Lyytinen, 2009; Jarke *et al*., 2009; Jarke and Lyytinen, 2010; Jarke *et al*., 2010): shorter time horizons; practically no greenfield IS development anymore; IT-supported ecosystems with exceedingly complex and non-linear dynamic dependencies between system components and their natural, technical, and social environment; or increased outsourcing and offshoring (Jarke and Lyytinen, 2010). Although many research results have found their way at least partially into practice (Hansen *et al*., 2008), many new challenges have arisen in the meantime, for example, increasingly distributed development teams (Hansen and Lyytinen, 2009). Even though agile approaches have shown positive effects, communication problems are still present in environments with many stakeholder groups and development teams (Pikkarainen *et al*., 2008; Vidgen and Wang, 2009; Lee and Xia, 2010). Purely informal communication may not be effective when dealing with distributed environments, a large number of stakeholders, and vast amounts of information in very complex, IT-supported ecosystems (Fitzgerald *et al*., 2006; Cao *et al*., 2009).

The fundamental issue of requirements development that we address – building consensus and shared understanding of requirements through social interactions – is still present in today's settings; it even has become more important. A major reason for this is the 'limits of individual cognition' (Hansen and Lyytinen, 2010) that stem from stakeholders' differing perspectives or 'frames of reference' (Bostrom and Heinen, 1977; Boland, 1979; McMaster and Grinder, 1980; Hansen and Lyytinen, 2010). It 'has repeatedly been observed that business and IT professionals "speak different languages" and apply differential yardsticks for desired outcomes' (Hansen and Lyytinen, 2010: 4). All stakeholders have to develop 'intersubjectively held mental models' (Gasson, 1999: 89) or 'shared technological frames' (Orlikowski and Gash, 1994) to reach consensus and build a shared understanding of requirements. However, not much research on ISD and requirements development goes beyond the impact of methods and principles (Kautz *et al*., 2007; Goeken and Patas, 2010; Siau *et al*., 2010). Few studies have actually investigated knowledge transfer, communication, and building of shared understanding (for example, Guinan and Scudder, 1989; Newman and Robey, 1992; Tan, 1994; Sawyer and Guinan, 1998; Watts Sussman and Guinan, 1999; Levina, 2005; Boh *et al*., 2007; Espinosa *et al*., 2007; Pikkarainen *et al*., 2008; Sawyer *et al*., 2010). There is a recognized need 'for theory and studies about social behavior and processes of communication, negotiation, and learning' (Kautz *et al*., 2007: 235).

ISD and requirements development as a language development process

In our research, we conceptualize requirements development as a *language development* and formalization *process* (Lyytinen, 1985: 61, italics added), where social interactions in ISD become visible when linguistic communication is observed. Communication using language is 'a form of social action constituted by social conventions for achieving social ends, premised on at least some shared understandings and shared purposes among users' (Tomasello, 2008: 343). It is the most striking form of social interaction and, at the same time, the most often ignored one because language is so natural and obvious to us (Tomasello, 2008: 12). We concentrate on the verbal use of natural language during requirements development. Natural language is always an issue in requirements development, at least in early stages, because of the heterogeneity of the involved stakeholders in usual ISD settings (Urquhart, 2001; Hansen and Rennecker, 2006). Regardless of the development approach used, at the beginning of the process the knowledge about the system is expressed verbally in natural language, even if in the long run formal documents supplement informal communication (Pohl, 1993: 281). This implies that the understanding of communication and the use of language are of critical importance and should be a fulcrum for research (Guinan and Bostrom, 1986; Urquhart, 2001).

Natural language's obvious advantage is that it is easier to comprehend and to access for a huge number of stakeholders (Berry and Kamsties, 2003). Nevertheless, there is a price for this: one of the most striking disadvantages is the inherent ambiguity of natural language because of the different 'frames of reference,' ambiguity that provides space for different interpretations of the same requirements. Several approaches exist that try to leverage the use of natural language for requirements development (Jarke and Pohl, 1993; Ambriola and Gervasi, 1997; Gervasi and Zowghi, 2005). These approaches mostly generate formal requirements out of natural language by using parsing algorithms, formal grammars, or similar techniques. We challenge this belief that a logical language can bring out and make explicit the complexities and subtleties of natural language (cf. a detailed discussion in Holten and Rosenkranz, 2011). Shared understanding is created and ambiguities in language are clarified, not by logical extraction, but by looking at (1) how the words or phrases in question are *used* in our daily activities and practices (Blair, 2005: 1), and (2) how they *represent* the meaning attached to them by the different stakeholders.

Some studies in IS research also investigate the relationship between language and IT systems (for example, Stamper, 1973; Kaasbøll, 1987; Auramäki *et al*., 1988; Holmqvist, 1989; Marakas and Elam, 1998; Gopal and Prasad, 2000; Stamper *et al*., 2000; Clarke, 2001; Urquhart, 2001; Alvarez and Urla, 2002; Wynn *et al*., 2002; Cohn *et al*., 2009; Pernille and Ojelanki, 2009; Beynon-Davies, 2010; Hansen and Rennecker, 2010; Holten and Rosenkranz, 2011). Specifically, the major focus on the study of language in IS research has been set by researchers using the Language-Action Perspective (LAP) (Goldkuhl and Lyytinen, 1982; Winograd and Flores, 1986; Flores *et al*., 1988;

Hirschheim *et al*., 1995: 202; Schoop, 2001). LAP is based on Speech Act Theory (SAT) (Austin, 1962; Searle, 1969) and makes use of linguistic communication to analyze and understand existing IT systems as well as to design new IT systems (Dwivedi *et al*., 2009). We share with LAP the assumption that *language use* can be observed as a social act in communication. However, LAP is mostly interested in 'the act of language rather than its representational role' (Winograd, 1988: 76), as a precondition for other actions (Hirschheim *et al*., 1995: 220–225). In contrast, we propose to investigate the creation of a shared language and its representation of meaning. Hence, we focus on the representational role of language, which corresponds to the act of reference or predication (Carnap, 1956: 6; Kamlah and Lorenzen, 1984: 19).

Semiotics and epistemology of the analysis of shared language and understanding

Recent cognitive studies show that languages fundamentally shape how we think and construct reality. Changing how people talk changes how they think, for example, language influences how we think about colour (Winawer *et al*., 2007), time (Boroditsky and Gaby, 2010), causality (Fausey and Boroditsky, 2011), or events, motion, emotion, and the contents of other minds (Boroditsky *et al*., 2010). Language and cognition clearly link to different mental 'frames of reference' and the building of shared understanding of stakeholders in ISD. For example, groups of people in the same organization carry out different functions and develop different 'jargons' or sub-languages on the basis of their professional backgrounds and the nature and organization of their functions (Nahapiet and Ghoshal, 1998; Weber and Camerer, 2003). As these languages form complex linguistic conventions, a 'word that denotes one thing in one context may even denote the opposite thing in a slightly different context' (Kaasbøll, 1987: 375). The development of a software-based IS will be influenced by these languages (Holmqvist, 1989: 73).

Human languages are symbolic languages (Deacon, 1997), and '[i]ndividual groups of humans develop their own unique ways of symbolizing and doing things – and these can be very different from the ways of other groups, even those living quite nearby' (Tomasello *et al*., 2005: 721). To analyze this specialization of languages and to explain how shared understanding of words – a shared language – emerges, we refer to semiotics, the study of signs. In semiotic theory and linguistics, a *sign* in a language is conceptualized as a combination of a 'meaning' or *concept* (the signified) and a 'representation' or *symbol* (the signifier) (de Saussure, 1974). Every language consisting of signs is based on conventions related to the *symbol-concept relationships*: any symbol-concept relationship is in essence an individual actor's (the speaker's or hearer's) interpretation, which is explained based on the semiotic tetrahedron (Falkenberg *et al*., 1998; Hesse *et al*., 2008) (cf. Figure 1), an adaptation of the semiotic triangle (Ogden and Richards, 1923). The semiotic triangle explains the relation between a *thing* (any object or entity from a domain), a *meaning*, and its *representation* along the three corners of the triangle. First, a *meaning* is derived from the interpretation of a thing (interpretation act). Second, the meaning is addressed using a *representation* of it (representation act). Thus, the relation from representation to thing is an indirect one, being dependent on an *actor's* interpretation of the thing (meaning). This leads to a fourth node in the centre of the triangle, essentially making it a tetrahedron.

From a semiotic point of view, symbols (for example, words or word groups that are used in oral utterances or written text) are representations and do exactly that: they represent *some* meaning. Then, to analyze the meaning a symbol obtains during the interaction between actors, we need to characterize its meaning in its specific linguistic context (Hoppenbrouwers and Weigand, 2000: 139). We focus in our research on the process that creates this relationship between symbols and meanings in social interactions. On the level of language use (pragmatics), meaning can be identified by analyzing the constitution of a semantic relationship (specific symbol-concept relationships) between utterances of interlocutors in conversations (Hoppenbrouwers and Weigand, 2000); a symbol acquires meaning by the *properties* ascribed to it by an actor (Kamlah and Lorenzen, 1984: 26–32).

Actors assign meanings (that is, concepts) to representations (that is, symbols) by predicating (Kamlah and Lorenzen, 1984), for example, by saying 'this is a computer' (and pointing at a computer) or by explaining the meaning by saying 'a computer is a programmable machine'. In the former example, the perception of an object of the 'real' world is used to assign meaning to a representation. In the latter example, people use already known and shared representations to create a new representation. Accordingly, the assignment of meaning can be *observed* by other interlocutors through monitoring the use of language. Epistemologically, we therefore assume that interlocutors can only create a shared understanding in the sense of inter-subjective understanding that is based on linguistic representations and their interpretation: the world can only be accessed through people's interpretations of it (Schutz, 1962; Lee, 1991: 347).

Although requirements are often based on existing entities (either 'real' material objects or abstract things), they are only accessible and can only be made inter-subjective

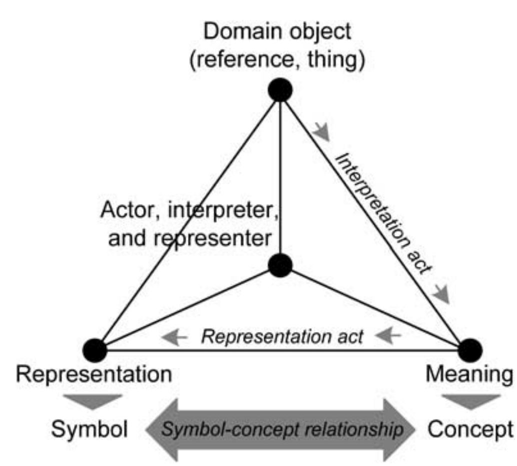


Figure 1 The Semiotic Tetrahedron (Falkenberg *et al*., 1998; Hesse *et al*., 2008).

by interlocutors 'taking the detour' and interpreting the other interlocutors' points of view. The interlocutors' points of view are transformed to 'a "know-how" held in common, the possession of a "language community"' (Kamlah and Lorenzen, 1984: 47) through interaction and communication. This linguistic communication is accessible for all interlocutors (and researchers that observe the linguistic communication; of course, this again is only an interpretation by the researchers). Thus, we understand 'concept' as the articulated part of 'meaning'. Although the latter is only in the mind and in the way people think, the former is manifested in the way people act and talk. Concepts are expressed in words and derive their meaning from the way they are used in daily life; rather than looking for abstract representations of meaning in the mind, from a linguistic point of view we look for patterns in the use of words and symbols (Tsoukas, 2005). The *linguistic or semantic alignment* of symbol-concept relationships during social interactions between humans is important for learning these linguistic conventions (Pask, 1975; Clark, 1996; Deacon, 1997; Brighton and Kirby, 2001; Smith, 2004; Deacon, 2005; Tomasello, 2008; Branigan *et al*., 2010). We investigate how stakeholders use language when developing requirements to align and build a shared language, and how this process leads to consensus and shared understanding of requirements between stakeholders. To do so, we build on a framework that we developed in previous research (Corvera Charaf *et al*., 2012). Accordingly, we assume that a concept can be unfolded stepwise by analyzing the linguistic context in which the symbol-concept relationship is defined and used in an ongoing language development process. Specifically, we examine the context of specific situations during requirements development by analyzing how symbols and concepts are introduced and negotiated, and how the symbol-concept relationships are collectively aligned and built during the interaction (Pickering and Garrod, 2004; Branigan *et al*., 2010). The following premise is given for our subsequent analyses:

> *Language Construct (LC) Premise*: The linguistic construction of a symbol-concept relationship can be observed during the language development process. The concept appears in its linguistic context and there may be several sides to the same concept. The symbol is used as a representation for given concepts. A symbol that represents one thing in one context may even denote the opposite thing in a slightly different context.

Consequently, we need to examine (a) how stakeholders use language (b) to develop a shared language as the basis for (c) a shared understanding of software-related and system-related requirements (Stamper *et al*., 2000). This leads to our first research question:

> **RQ1**: How does language use give rise to a shared understanding between stakeholders in requirements development ('the usage question')?

Communication should also be more efficient once shared conventions and a shared language have emerged (Moldoveanu, 2002; Weber and Camerer, 2003; Selten and Warglien, 2007). However, this efficiency of a shared language requires that representations (utterances, symbols) be *effective* in the first place, that is, always anchored to the unique and particular situation of their use (because the significance of a representation depends on the concrete circumstances and context in which the representation is used, Hirschheim *et al*., 1995: 157). That is why we also investigate the so-called 'representation question (how well is the result represented?)' (Hirschheim *et al*., 1995: 144):

> **RQ2**: How is shared understanding captured in representations, and how are coherent and meaningful representations constructed and sustained over the whole requirements development process ('the representation question')?

Language quality in the language development process

The concept of language quality: how to address the quality of representations

We suggest applying the concept of quality to analyze how shared understanding emerges from language use and is carried on during the requirements development process. *Quality* is often understood as a degree of excellence and has been equated with 'performance' and 'effectiveness' or 'efficiency' and 'productivity'; measurement is largely erratic (cf. Shenhav *et al*., 1994 for a detailed discussion in organization studies). Quality may also be broadly defined as the discrepancy between one's expectation of a thing's characteristics and one's actual experience of its characteristics – perceived bad quality results from an absence of expected characteristics (Anupindi *et al*., 2006: 245). We understand quality in this sense, as an expression of the intuitively evaluated 'goodness' of a thing, which requires some definition of expected characteristics and a measure for them.

As our research focuses on the emergence of shared language and shared understanding of stakeholders during requirements development, we need to translate the concept of quality into a definition applicable to the language development process. We define *language quality* in requirements development by focusing on how well a representation captures and retains a meaning assigned to it, complying thereby with a stakeholder's expectations:

> *Language Quality (LQ)* is the degree to which a symbol obtains and retains a relationship to a concept.

This definition is directly linked to RQ2 and the 'representation question' (Hirschheim *et al*., 1995: 144); representations that represent an object most appropriately and retain this meaning over time are more effective (that is, 'have higher language quality'). Therefore, we assert that, in order to measure and evaluate language quality in the language development process, we need to focus on how well the symbol-concept relationship (addressed in the LC Premise) is built:

> *Language Quality (LQ) Premise*: The quality of the built language can be observed as mappings between given

symbols and concepts in the context of a language development process.

'Effectiveness' in requirements development, then, refers to the degree to which interlocutors manage to achieve language quality. This conceptualization of language quality focuses on one specific aspect of language in relation to social interactions – the question of how shared understanding and meanings emerge. Although we argue that shared understanding emerges as a result of social interactions, of course, other events and social interactions may also lead to its dissipation, for example, a change in project staff and in particular in project leadership.

Measuring the dimensions of language quality

We focus on how structures in people's minds are represented on the observable level of language use. We are, first, interested in measuring how a representation becomes a 'good' representation from an interlocutor's point of view and, second, how it remains one throughout time. Language is the venue where we can observe how the shared understanding of domain objects is being built during the interaction of interlocutors. Our goal is to go to the venue where the representation is negotiated and aligned through language use.

In order to assess and measure language quality, we adapt Wand and Wang's (1996) articulation of a minimal and consistent set of dimensions for the concept of data quality and transfer these dimensions into our setting to measure language quality from an interlocutor's point of view. Wand and Wang (1996) describe *data quality* based on the notion that the role of an IS in to provide 'a representation of an application domain [...] as perceived by the user', where the user's view is a 'standard against which data quality is defined' (p. 88). Guided by this definition, Wand and Wang (1996) propose four dimensions of data quality: *completeness*, *meaningfulness*, *non-redundancy*, and *unambiguity*. These dimensions act as classification criteria for the quality of representations and result from the mapping between a user's interpretation and a given representation. They were developed by observing deficiencies during this mapping and deriving therefrom criteria for 'good representations' (Wand and Weber, 1995: 209). From a user's point of view, a given *representation is good* if it is *complete, meaningful, non-redundant, and unambiguous*. Hence, building on our LC Premise, determining the goodness of representations in our case of *language quality* corresponds to comparing (a) the articulation of an interlocutor's understanding of symbol-concept relationships with (b) the articulation of other interlocutor's understanding of symbol-concept relationships. Therefore, to assess the mappings between symbols and concepts in a language development process (LQ Premise), we adapt the quality dimensions provided by Wand and Wang (1996). We define language quality along four *dimensions of language quality* (cf. Table 1).

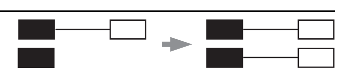
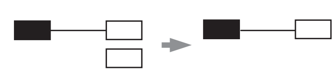
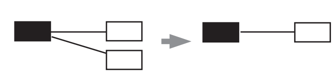
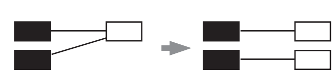
We note that our dimensions of language quality are consistent with semiotic theory and the epistemological background discussed earlier. Although Wand and Wang (1996) base their dimensions on a different epistemological background, their understanding that good data quality relates to 'good representations' from the point of view of a user (Wand and Wang, 1996: 88) is analogous to our understanding of language quality that focuses on how well a representation captures and retains a meaning assigned to it, complying thereby with a stakeholder's expectations.

Research study

Study design and data collection

Our analysis of language quality during requirements development concentrates on the language development process that consists of linguistic utterances with language-defining character (that is, symbols and concepts are introduced and negotiated by stakeholders). In order to use 'naturally occurring data' (Silverman, 1998: 18), we opted for collecting data within an observational field study, where we observed the phase of requirements development in a real ISD project. We inspect the linguistic utterances spoken by stakeholders during requirements development in the selected project. The project developed an internet-based content management system (CMS) and was

Table 1 Criteria for good representations: Dimensions of language quality

Quality dimension	*Description*	*Repair action*
Completeness	Completeness is given when a representing symbol is given for every concept.	
Meaningfulness	A meaningful representation is given when every representing symbol is linked with one corresponding concept.	
Non-redundancy	Non-redundancy is given when in every linguistic unit, single concepts are linked to one concrete symbol.	
Unambiguity	Unambiguity is given when no two concepts map into the same symbol.	

Lengend: concept symbol

Source: Adapted from Wand and Wang, 1996.

conducted from December 2006 to September 2009.[1] A traditional, plan-driven development approach was used. The common, natural language spoken in the project was German. (Note: all reported utterances in this article are translated by the authors.) The project team consisted of three people from a project management team (PM), six from an enterprise team (EN), five from a developer team (DE), and two from a requirements engineering team (RE). The second author was also a project member and part of RE (denoted by speaker ID 'RE2' in examples). She was involved as an 'observing participant' (Alvesson, 2003: 174), with none of her actions purposely influencing the research besides her project-related tasks, making notes, and operating the recorder. We had access to most of the project meetings that took place in different stakeholder configurations. We did not provide any kind of feedback to the other participants during the course of the project to avoid any kind of influence on the natural course of social interactions. We were able to include in our analysis data collected in nine project meetings between December 2006 and January 2009. The first four meetings were observed by the participating researcher, and the observations of the language development process were captured as field notes. For the following five meetings, the authentic communication was audio recorded and transcribed. This resulted in about 15 h of audio recordings that were transformed into 238 transcript pages with a total of about 107,800 words.

Tracing the dimensions of language quality during language use: coding categories

We scrutinized all the collected data, looking for single concepts that were relevant to the project context or situation. We looked for sequences where a relationship between symbols and concepts is built. All sequences corresponding to a particular concept were subsumed under the same *object of definition* (OD). One OD comprises one or many sequences from different meetings where either a (new) symbol has been added or the concept has been amended. We could identify 70 ODs from our data for our analysis (cf. Appendix A for an example). In order to ensure logical consistency and adequacy, we deduced coding categories that are in line with the theoretical premises of our research (Schutz, 1962). The identification of symbols and concepts occurred along a predefined coding category based on the LC Premise. The resulting *LC Category* is our first coding category and consists of the two sub-categories 'concept' and 'symbol' (cf. Table 2; Appendix A gives detailed coding rules).

Note that the question of '*what* concept is being represented' is solved by the interlocutors themselves. We do not ask whether a domain object behind the concept is a material object, an abstract entity, or a virtual or even fictitious object (Falkenberg *et al*., 1998: 46). Furthermore, as we only observed and did not audio record some meetings and because stakeholders also communicated using e-mail or telephone, our observations do not include the introduction of all symbols and concepts involved. The detailed analysis of sequences centers on the audio-recorded meetings.

The second coding category is derived from the LQ Premise. According to the rationale of analyzing the symbol-concept relationships, our analysis of language quality builds on the identification of deficiencies with regard to the dimensions of language quality described in Table 1. The resulting sub-categories of the *LQ Category* ('incompleteness', 'meaninglessness', 'redundancy', and 'ambiguity') are the basis for the codes (Table 3).

Two coders performed the coding following the coding procedure that is illustrated in Figure 2. The coding along the LC Category was performed first (*LC Coding*). During the LC Coding, utterances were classified as SY (symbols) or CO (concepts) depending on the corresponding OD. This provided the basis for the more detailed second coding based on the LQ Category (*LQ Coding*). During the LQ Coding, the relation between utterances that were classified as SY and CO was analyzed in the following way: If the relation between SY and CO could be identified in the observed sequence, codes INC (inconsistency) or MLN (meaninglessness) were used to encode the appropriate case. INC was assigned if for a given CO1, no SY1 could be identified. MLN was assigned if no CO1 was introduced for SY1. In the other cases (concepts with assigned symbols, for example, SY1 and CO1), the identified relations were further analyzed using codes AMB (ambiguity) and RDC (redundancy): the RDC code was assigned if another symbol (SY2) also referred to CO1; the AMB code was used if another concept (CO2) also referred to SY1.

The evaluation of inter-coder agreement for the complete LC Coding yielded an inter-coder reliability of 0.84 (Cohen's kappa, Landis and Koch, 1977). The evaluation for the LQ Coding focused on a subset of the data of 20%, which yielded an inter-coder reliability of 0.83. After LC Coding and LQ Coding had been conducted, each coding was assigned a unique *coding ID*. The coding ID consists of the OD number (1–70, cf. Appendix B), the code of the LQ sub-category ('I' = INC, 'M' = MLN, 'R' = RDC, 'A' = AMB, cf. Table 3), and the number of the coding in

Table 2 Language Construct (LC) category

Sub-category	*LC code*	*Coding description*	*Examples*[a]
Concept	CO	Noun or set of nouns introducing new intrinsic or mutual properties.	…[*this is a kind of questionnaire at the end of every phase*]…
Symbol	SY	Noun or set of nouns acting as a substantial representation of the concept meant. Stands for and represents the concept.	There are [PHASES], which … There are [SUBSCRIBED MEMBERS], which…

[a]In quotes, concepts are highlighted in italics and symbols are highlighted in small caps.

Table 3 Language Quality (LQ) category

Sub-category	*LQ code*	*Coding description*	*Coding examples*[a]
Incompleteness	INC	After having described a concept, no symbol is introduced or determined to refer to this concept.	**Quotes** / **LC Code** / **LQ Code** A: Is there [*something like in Wikipedia, where in the middle of the text you have blue text for relevant words, then you can click on it and it automatically leads you to (…)*]? — CO — INC B: Yes.
Meaninglessness	MLN	After having used a symbol, no definition of the symbol is given and the subsequent utterances make clear that the corresponding concept is missing or unclear.	**Quotes** / **LC Code** / **LQ Code** A: The [USER]… — SY — MLN B: What do you mean with [USER]?
Redundancy	RDC	After having used a symbol for the reference of a concept, a different symbol is used for the reference of the same concept.	**Quotes** / **LC Code** / **LQ Code** A: There are [SUBSCRIBED MEMBERS], which… — SY (…) B: The [REGISTERED USERS]… — SY — RDC
Ambiguity	AMB	After having used a symbol for the reference of a concept, a different concept can be detected, which is represented by the same symbol.	**Quotes** / **LC Code** / **LQ Code** A: … As I understand it, it is only for you, so that you can [*set up your*] [PROFILE]... — CO SY B: No, I mean specifically the [CONTACTS], not the [PROFILES]. — AMB

[a]In quotes, concepts are highlighted in italics and symbols are highlighted in small caps.

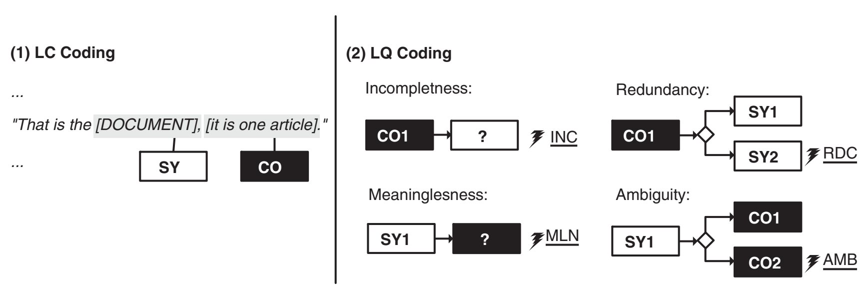


Figure 2 Coding Procedure for LC coding and LQ coding.

the order of occurrence for each OD and LQ sub-category (cf. Appendix C). Subsequently, we interpreted and analyzed every OD individually in order to identify reasons why language quality changed. For every dimension of language quality (cf. Table 1 and Table 3), we analyzed the elements that had a positive or negative impact on language quality. This finally led us to derive a set of propositions, which were gained by our explorative analysis of the codes and codings (induction).

Analysis of language quality during requirements development

Overview

Of the 70 ODs in total, we observed 53 ODs with a total of 126 deficiencies related to the language quality dimensions of the LQ Coding. Table 4 summarizes the identified deficiencies for each dimension. (Note that groups of ODs with deficiencies (second column) are not disjunctive.)

Our analysis and interpretation of the data revealed different insights about the dimensions of language quality, which are summarized in the set of propositions given in Table 5. In the following section, we give an overview of the emergence of these propositions by presenting the results of our analysis for all four-language quality dimensions. We also present three specific issues that are related to more than one dimension. To link results to the propositions, we include an arrow and the corresponding proposition number in brackets (for example, '($\rightarrow$P1)'). (The respective ODs with deficiencies and codings for every subset are given in Appendix D.)

Deficiencies in completeness

We analyzed whether and how a representing symbol was assigned to an existing concept for all 29 cases of a deficiency in completeness. Departing from a categorization of the representing symbol, we discerned three subsets:

- *Subset INC-1* includes 20 cases of deficiency where, subsequently, a specific symbol was attached to the given concept (for example, symbols such as '[AUTOMATIC TAGGING]' or '[PERSONALIZATION CONCEPT]').
- *Subset INC-2* consists of four cases where no specific symbol was introduced at all or symbols were abolished (for example, concepts such as [*…set a checkmark 'this document is a method description' yes/no.*]).
- *Subset INC-3* refers to five cases where no specific symbol was attached to the given concept but a descriptive symbol was used, consisting of single words of an already uttered description (for example, '[THE DISPLAY THING]' instead of '[RSS-FEED]' or '[THE ISSUE ABOUT ROLLING-OUT]' instead of '[TOGGLE FUNCTION]'). (For the analysis of subset INC-3, cf. section 'Multidimensional Issues'.)

Among the cases of subset INC-1, we first paid attention to the number of actions it took to introduce the corresponding symbol and identified 13 cases in which completeness was achieved after just one action (subset INC-1.1) The most evident example of a purposeful action to achieve completeness included an explicit request for a symbol:

OD-40 'Morphological Box' (Coding 40-I1)	
Speaker	*Linguistic Utterance*
DE3:	I think about this… what is it called? [(*description of the concept*)].
RE1:	[morphological box]

Here, the requesting speaker was aware that a specific symbol for referring to the concept existed. In four other cases (coding IDs 4-I1, 23-I1, 42-I1, 58-I1), although no one purposefully requested a symbol, corresponding symbols were spontaneously included in the feedback in a self-evident way. For example, in OD-4:

OD-4 'Automatic Tagging' (Coding 4-I1)	
Speaker	*Linguistic Utterance*
PM2:	Is it possible to [(*description of the concept*)]?
DE3:	[Automatic tagging] is too difficult.

Although the previously mentioned cases of subset INC-1.1 are self-evident concerning the introduction of corresponding symbols, all remaining eight cases (coding IDs 6-I1, 27-I1, 28-I1, 29-I1, 30-I1, 32-I1, 50-I1, 61-I1), rather than being self-evident, show uncertainty concerning the adequacy of the introduced symbols:

OD-6: 'Bread Crumbs' (Coding 6-I1)	
Speaker	*Linguistic Utterance*
DE3:	Eh… just a little bit something like… eh… such [BREAD CRUMBS]?

'Uncertainty' here means that stakeholders are uncertain about the specific meaning of symbols that are used during conversations, and about whether they have the same understanding of a symbol as other stakeholders (that is, if shared understanding has indeed been achieved), because the conveyed meanings are perceived as vague, unstable, or subject to change. This indicates that even if the achievement of shared understanding occurs, it is in several cases a process fraught with uncertainty, where the stakeholders are not sure whether or not they have actually achieved shared understanding (→P1; cf. also the discussion of subsets INC-3 and RDC-3 in the section 'Multidimensional Issues').

The remaining seven cases are characterized by showing several actions before the achievement of completeness (subset INC-1.2). They share with the four cases of subset INC-2 the fact that at least the first actions after introducing the concept did not yield new symbols to represent it.

Table 4 Summary of deficiencies

LQ Sub-Category	*Number of ODs with deficiencies*	*Number of deficiencies*
Incompleteness	25	29
Meaninglessness	15	15
Redundancy	24	65
Ambiguity	10	17
Total	53 (out of 70 ODs)	126

Table 5 Propositions

Dimensions	*No.*	*Proposition*
Completeness	P1	The achievement of shared understanding by language development is a process fraught with uncertainty.
Completeness	P2	The absence of representing symbols for relevant concepts increases interaction time.
Completeness, meaningfulness, Non-redundancy, unambiguity	P3	Explicitness in the request for the definition of symbols and/or concepts increases the possibility of achieving shared understanding of requirements.
Completeness, Meaningfulness, Non-Redundancy, Unambiguity	P4	Explicitness in the request for the definition of symbols and/or concepts is not self evident.
Completeness, Non-Redundancy	P5	The use of volatile symbols hinders the achievement of language quality, which in turn reduces the effectiveness of requirements development.
Non-Redundancy, Unambiguity	P6	The existence of partially similar symbol-concept relationships requires a higher consideration of language quality by involved stakeholders.

Although in six of these cases the reason can be attributed to the further definition of the given concept, in five cases (coding IDs 17-I1, 37-I1, 45-I1, 55-I1, 63-I1) the subsequent actions consisted of repeating an already given concept instead of using a symbol. For example, for OD-17:

OD-17 'Embedded Hyperlink' (Coding 17-I1)	
Speaker	*Linguistic Utterance*
PM2:	Is there [*something like Wikipedia, where in the middle of the text you have blue text for relevant words and then you can click on it and it automatically leads you to another module* (…)]?
RE2:	Yes.
PM2:	I mean [*so that you can jump immediately to another topic…*]
RE1:	I do not know if that happens automatically.
PM2:	No, [*I mean…*]

Here, PM2 tries to explain the concept that is represented by the symbol '[EMBEDDED HYPERLINK]'. Although he already received an answer to his question after the first run, he explained the concept again and again to ensure that he and RE1 were talking about the same concept. This supports P1, which argues that the absence of representing symbols (that is, specific, designated symbols instead of mere descriptive statements) for relevant concepts leads to stakeholders' being uncertain about whether shared understanding has been achieved. Moreover, as the repetition of already given concepts is a time-consuming activity, it increases the overall interaction time (→P2).

Deficiencies in meaningfulness

In all 15 cases where we observed that a symbol was introduced, we analyzed the subsequent interaction to identify deficiencies in meaningfulness and to examine how these deficiencies were addressed. From then on, we analyzed whether and by which actions the 'missing' meaning was adjusted or negotiated. Among all identified cases, we discerned two main courses of action:

- *Subset MLN-1* comprises seven cases in which stakeholders gave definitions of terms spontaneously without others requesting them.
- *Subset MLN-2* lists five cases with explicit requests for the meaning of symbols that had been already introduced.

It is possible to identify and analyze deficiencies in meaningfulness only if the deficiencies are expressed by the interlocutors during the interaction. For the cases of subset MLN-1, the meaning-adding action depends on the speaker's own recognition or estimation of the meaninglessness of a symbol for the current audience. For example, in a sequence for OD-2:

OD-2 'Anchor' (Coding 2-M1)	
Speaker	*Linguistic Utterance*
DE3:	I am thinking about [ANCHORS]. These are […]

Among the cases of subset MLN-2, a shared understanding was triggered by an explicit request for a concept. For example, in OD-35:

OD-35 'Method Description' (Coding 35-M1)	
Speaker	*Linguistic Utterance*
RE1:	Links to [METHOD DESCRIPTIONS]
DE3:	What does that mean?

In four of the cases (coding IDs 15-M1, 23-M1, 25-M1, 35-M1), the uttered request appeared as specific enough as it yielded the articulation of the concept as an answer. A different case is OD-36. Figure 3 presents the transcript and coding for the corresponding sequence in more detail.

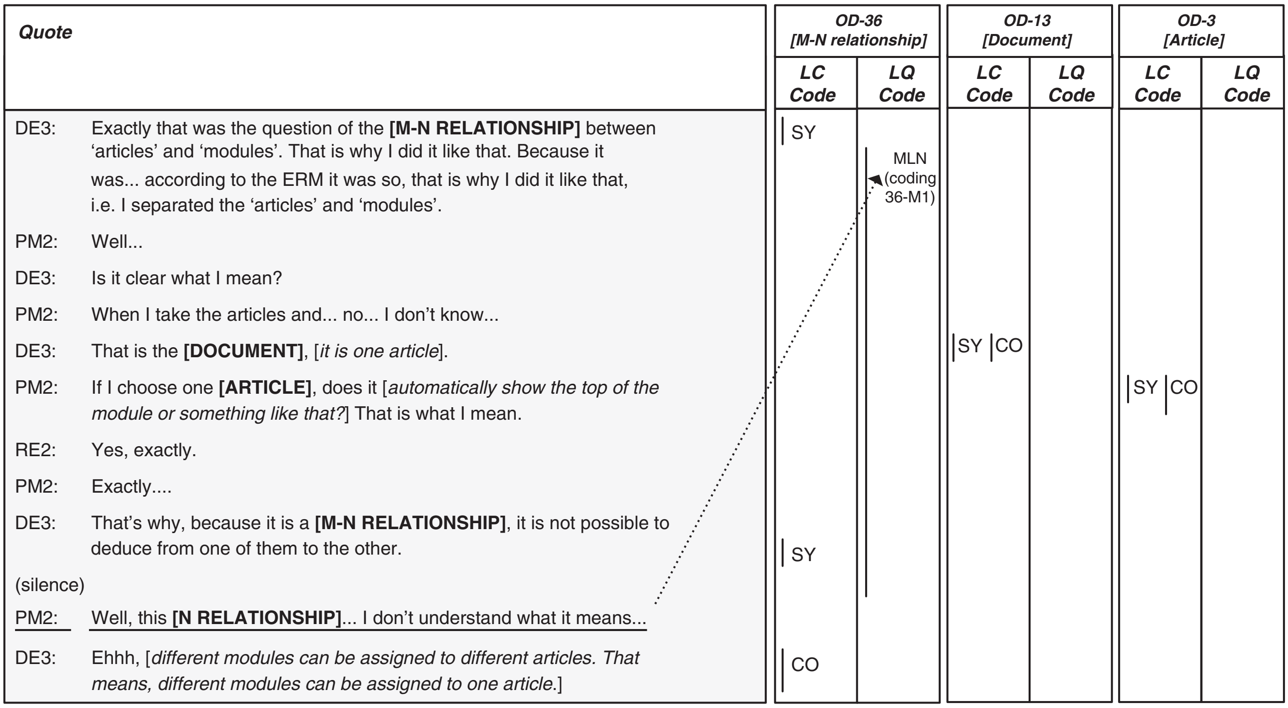
Here, the concept addressed by the symbol '[M-N RELATIONSHIP]' was not understood by one of the stakeholders (PM2). Although he clearly expressed that there was a misunderstanding, he did not explicitly articulate the reason for his confusion. This led to the definition of OD-3 and OD-13 – (Article) and (Document) – by both the involved stakeholders (DE3 and RE2). After some discussion, PM2 finally expressed explicitly that the symbol '[M-N RELATIONSHIP]' was unclear to him (underlined sentence in Figure 3). At this point, deficiencies because of meaninglessness could be identified (as indicated by the dotted arrow). As this case suggests, the meaningless symbol is not always the first subject of discussion when it comes to misunderstandings because of unspecified requests.

To sum up, all cases of subset MLN-2 indicate that if stakeholders explicitly request the definition of symbols and/or concepts, then this positively influences the achievement of shared understanding (→P3). Moreover, OD-36 suggests that the given request needs to be precise and definite to have this impact. Three of the 15 ODs for meaninglessness (33, 53, 64) could not be classified to either course of action, for example, OD-33, where the concept for the symbol '[KNOWLEDGE PLATFORM]' was given, but in the sense of reassuring the own notion of its meaning.

Deficiencies in non-redundancy

Deficiencies in non-redundancy were observed in 24 ODs. In each of the ODs, we identified one or more additional symbols that mapped to the same concept, yielding a total of 65 cases of deficiency, which constitutes the highest number of codings among all dimensions and represents more than 50% of the overall deficiencies in language quality. Similar to deficiencies in completeness, the analysis departs from the categorization of the symbols involved in the cases. We discerned the following subsets:

- *Subset RDC-1* comprises 26 cases and refers to the use of specific, redundant nouns, that is, synonyms standing for the same concept (for example, '[NAVIGATION PATH]' instead of '[BREAD CRUMBS]).
- *Subset RDC-2* (23 cases) focuses on specific, redundant nouns derived from part-similar concepts or from a shortened or generalized version of a previously given



Legend: SY: Symbol; CO: Concept; MLN: Meaninglessness

Figure 3 Excerpt of Coding for OD-3 (Article), OD-13 (Document), and OD-36 (M-N-Relationship).

symbol (for example, '[USER]' instead of '[REGISTERED USER]').

- *Subset RDC-3* (16 cases) refers to the use of no specific symbol but a descriptive symbol (for example, '[PHASES-MODUL-THING]' instead of '[PROCEDURE MODEL]'). For the analysis of subset RDC-3, see section 'Multidimensional Issues' (cf. also the remarks for subset INC-3 in section 'Incompleteness' above).

An example for subset RDC-1 can be found within OD-19. In this case we observed discrepancies resulting from the use of the symbols '[FIELD REPORT]' and '[CASE STUDY]' by different speakers to refer to the same concept. At an earlier stage of the project, the stakeholders agreed on integrating other companies' experiences as a particular document type in the CMS. Although all stakeholders seemed to link both symbols '[FIELD REPORT]' and '[CASE STUDY]' to this same concept, the parallel use of both symbols apparently brought confusion, as the stakeholders showed signs of uncertainty, for example, by beginning a sentence using one symbol and ending it with the other. Nevertheless, there was no explicit request to adopt one of the two symbols (→P4). Such requests are therefore not always self-evident if they are not explicitly expressed, and both the explicitness and the recognition of related deficiencies by the involved interlocutors cannot be taken for granted.

It took two meetings until everyone implicitly seemed to have adopted the use of '[FIELD REPORT]'. Then, in the ninth meeting, a third, new symbol was introduced because one of the project documentations addressed '[FIELD REPORT]' as '[EXPERIENCE REPORT]'. Apparently motivated by a third redundant symbol referring to the same concept, one of the stakeholders emphasized the redundancy by explaining:

OD-19 'Field Report' (Coding 19-R1)	
Speaker	*Linguistic Utterance*
RE1:	In the ERM we called it [EXPERIENCE REPORT] instead of [FIELD REPORT]. The terms are not really consistent.

In the following utterances he addressed the concept as '[CASE STUDY/EXPERIENCE REPORT]', until after some discussion rounds he switched to just '[FIELD REPORT]'.

Although the deficiencies of subset RDC-1 seem to derive from people not using the same symbol consistently, the difficulties of subset RDC-2 can also be interpreted as stemming from the inaccurate use of symbols. Subset RDC-2 includes several deficiencies in unambiguity resulting from the additional use of either shortened or generalized versions of the original symbols. For example, OD-67 shows two cases of redundancy, as the symbol '[USER PROFILE]' was replaced by the symbols '[OPERATOR PROFILE]' (coding ID 67-R1) and '[PROFILE]' (coding ID 67-R2). For these cases, the coding of redundancy shows also coding of ambiguity, as these redundant symbols are linked with symbols of other ODs. Therefore, we discuss this topic in the section 'Multidimensional Issues'.

Deficiencies in unambiguity

With regard to deficiencies in unambiguity, we analyzed whether and by which actions the ambiguous meaning was adjusted or negotiated. The analysis focused on two types of cases, subsumed as follows:

- *Subset AMB-1* comprises five cases in which we could recognize whether the given symbol stood for a second concept by investigating the *subsequent utterances*.
- *Subset AMB-2* focused on 12 cases in which the stakeholders' utterances did not address ambiguity in the first place. Instead, we analyzed the used symbols and compared them with symbols of *other symbol-concept combinations* in all meetings of the project.

Concerning subset AMB-1, in three of the cases (coding IDs 10-A3, 19-A1, 23-A1) the clarifying action was performed by the same stakeholder, as for example in this sequence for OD-23:

OD-23 'H1 Tag' (Coding 23-A1)

Speaker	*Linguistic Utterance*
RE1:	There you have a [*header*] [H1]. Well, *not* a [H1 Engine], but a [H1] [*HTML tag 'method descriptions' and there…*]

In two other cases (coding IDs 10-A2, 49-A1) the stakeholders who were not using the ambiguous symbol requested clarification, enabling a shared understanding. For example, in OD-10, [CONTACT OBJECTS], the stakeholder explained his understanding behind the given symbol with a description of the concept for OD-67, [USER PROFILE]):

OD-10 'Contact Objects' (Coding 10-A2)

Speaker	*Linguistic Utterance*
PM3:	What do you mean by that? There with [CONTACT DATA]. I thought [[*concept for OD-67*]].

In addition to the deficiencies explicitly pointed out by stakeholders, the analysis of the interrelatedness of symbols and concepts across the ODs helped us to identify several other deficiencies in unambiguity (subset AMB-2). These are also explained in the following section.

Multidimensional Issues

The contingency of language quality (incompleteness, redundancy)

Among the deficiencies related to symbols (dimensions of completeness and non-redundancy), we identified several cases where, instead of a specific and concrete symbol a descriptive symbol was added (subsets INC-3 and RDC-3). For example, subset INC-3 shows cases in which no initial symbol for a concept was explicitly defined. Therefore, when the stakeholders needed to refer back to these concepts that had already been discussed, they helped themselves by using single words of the already uttered description. These descriptive symbols were of a volatile nature. We subsequently refer to 'volatile symbols' in cases when, unlike the cases of specific or designated symbols, symbols are nonpermanent for a concept in the sense that they are seldom reused or even completely abandoned. Specifically, this volatile nature adds to the perceived uncertainty and, in turn, occurs when stakeholders are uncertain if a symbol is adequate to convey a specific meaning (→P1). For example, in a sequence for OD-31 the stakeholders were discussing either storing genuine, self-generated data in the CMS or just displaying the data of other news platforms:

OD-31 'Integration of News 3' (Coding 31-I2)

Speaker	*Linguistic Utterance*
DE3:	In terms of practicability I would rather [*forward directly to other websites*]. (…)
RE1:	I also think that the platform can be understood as [*a display for other's news*].
(…)	
PM2:	I also think so, the [DISPLAY THING] is better.

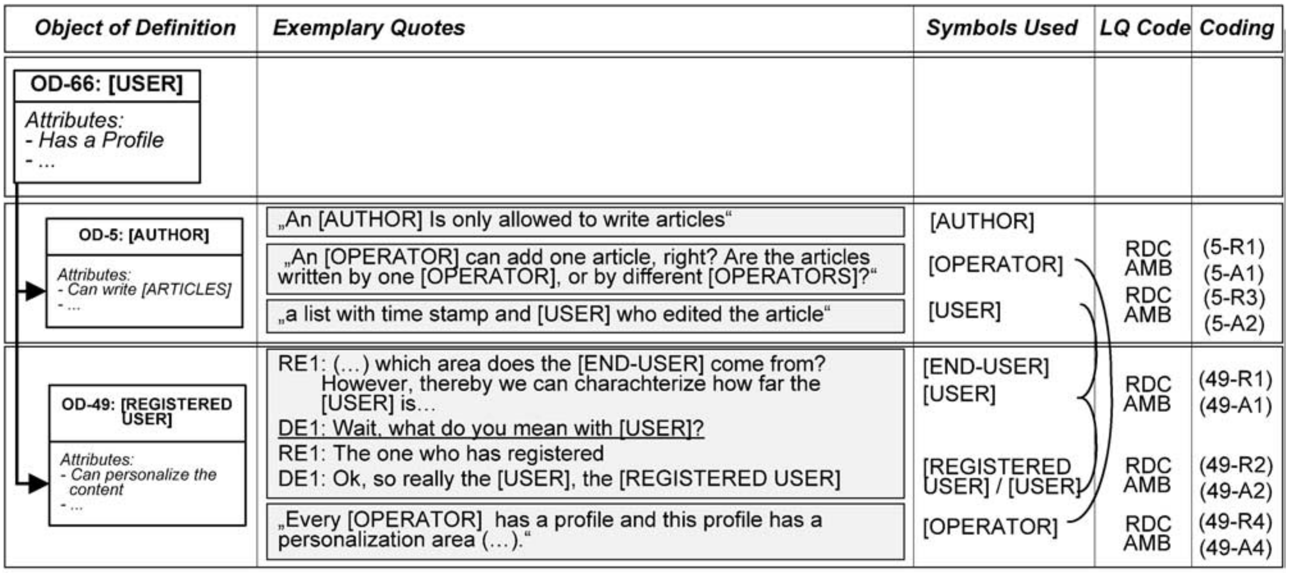
Similar observations can be identified for the cases of subset RDC-3.

Concerning the building of shared understanding (RQ1), the function of descriptive symbols seems to be appropriate: they *somehow* convey meaning. This seems doubtful for the question of effectiveness of requirements development with regard to language quality, that is, whether a symbol retains a concept addressed to it (RQ2). The reason is that these symbols are very situation specific and depend greatly on the context. They obtain meaning in relation to previously given descriptions and their representational character is of a volatile nature (→P5). For example, in the observed cases not one of these symbols was used beyond the situation of its emergence. Moreover, although they fulfill the task of a representation in a specific situation, their use may impede an agreement on a representing symbol in the long term, as for example in OD 68 (subset RDC-3), where the primarily used symbol '[VISUALIZATION]' was abolished by the descriptive, volatile symbols '[OVERVIEW GRAPHIC]' (coding 68-R1) and '[CHART]' (coding 68-R2), leading to deficiencies of redundancy.

The interrelatedness of requirements (redundancy, ambiguity)

We identified 14 cases for the dimension non-redundancy in which the redundant symbol of a specific OD was at the same time used as a symbol for another OD, fulfilling the assertions about deficiency in unambiguity. For example, the concept of [User] had been defined as a general class of objects, referred to by the symbol '[USER]'. In addition, various specializations of this general class, such as [Administrator], [Author] or [Registered user], were initially addressed by corresponding symbols such as '[ADMINISTRATOR]', '[AUTHOR]', and '[REGISTERED USER]'. The analysis of these ODs revealed several deficiencies of redundancy and ambiguity. Figure 4 shows an excerpt of the data and corresponding codings.

Concerning OD-5, the first exemplary case shows the use of the symbol '[AUTHOR]' with the concept of OD-5, [Author]. The two following exemplary cases show excerpts



Legend: AMB: Ambiguity; RDC: Redundancy

Figure 4 Excerpt of Coding for OD-5 (Author), OD-49 (Registered user), and OD-66 (User).

of the discussion about editing access rights and about a list of all revisions of an article. There, the stakeholders referred to OD-5 using the symbols ‘[OPERATOR]’ and ‘[USER]’, causing redundancy and ambiguity at the same time (coding IDs 5-R1, 5-A1, 5-R3, 5-A2). In a similar way, the concept for OD-49 [Registered User] was addressed by different symbols: ‘[END-USER]’, ‘[USER]’, ‘[REGISTERED USER]’, and ‘[OPERATOR]’. In only one of these cases could we identify a clarifying action, as RE1 requested explicitly the definition of the symbol ‘[USER]’ (underlined sentence). Nevertheless, we observed no explicit request for the adoption of just one non-redundant and unambiguous symbol in the entirety of OD-5, OD-49, and OD-66.

However, the (initially) undiscovered ambiguity between related concepts and symbols can be accounted for by the fact that these related concepts have similar attributes. Cases dealing with generalization and specialization have similar or related symbol-concept combinations and may be more susceptible to producing deficiencies in unambiguity and non-redundancy. Moreover, these deficiencies also affect other ODs that build on the same symbols. For example, the redundant use of the symbols ‘[USER PROFILE]’ and ‘[OPERATOR PROFILE]’ in OD-67 suggests that the source is the redundancy within OD-66, [USER]. On the other hand, the similarity between concepts can provide misleading evidence that everyone is referring to the same concept. The risk of remaining undetected can be interpreted as being higher. Therefore, we assume that this is especially relevant to the field of ISD, as the differentiation between superclass and subclass is a fundamental principle often used in ISD. This leads to partially similar symbol-concept relationships that may cause lower language quality, as seen in the examples above – ‘[USER]’ being used as a symbol for both the concepts [OPERATOR] and [END-USER]. Hence, we propose that the existence of partially similar symbol-concept relationships requires a higher consideration of language quality (→P6). We need to explicitly pay attention to distinctions between similar symbol-concept relationships in language use to avoid redundancy and ambiguity.

Taking action for improving language quality (all four dimensions)

Lastly, we analyzed cases where we interpreted that stakeholders explicitly triggered the achievement of a shared understanding. Among the deficiencies related to concepts (dimensions meaningfulness and unambiguity), we identified an explicit request for a meaningful concept in five of 15 ODs and an explicit request for the clarification of ambiguity in two of 10 ODs. Among the deficiencies related to symbols (dimensions of completeness and non-redundancy), we identified an explicit request for a missing symbol in one of 25 ODs and an explicit signal of redundancy in six of 24 ODs. These kinds of purposeful actions are treated in prior literature under the topic of *communicative breakdowns* (Milroy, 1984; Winograd, 1988), where stakeholders react to some kind of unreadiness, aiming for successful communication. In all observed cases, such a breakdown led to the achievement of shared understanding (→P3). Specifically, however, the ODs with deficiencies in redundancy and ambiguity show the need for a proactive intervention to improve language quality: in 18 of 24 ODs and five of 10 ODS, respectively, deficiencies have neither been explicitly taken into account nor explicitly solved by the stakeholders.

Discussion

Our research examined the building of shared understanding of requirements. As the literature states a need for studies about social behavior and processes of communication, negotiation, and learning, our primary goal was to open at least in part the black box of the ISD process. We therefore investigated social interactions during requirements development as a process of language development and analyzed how stakeholders use language to build and align a shared language as the basis for the shared understanding of requirements. Led by the questions of how a shared understanding between stakeholders emerges through language use (RQ1) and how it is captured in linguistic representations over the whole requirements development process (RQ2) within the project team, we proposed the concept of language quality for evaluating the language development process and the goodness of shared language in the context of an ISD project.

The dimensions of language quality – completeness, meaningfulness, non-redundancy, and unambiguity – provide a coherent set of criteria to analyze the process of building a shared understanding of requirements. The application of these dimensions to real communication data of a requirements development process in an ISD project was preceded by a translation of our two theoretical premises (LC Premise and LQ Premise) into suitable research categories of a coding scheme (LC and LQ Category). By applying the concept of language quality to recorded and transcribed conversations from requirements meetings of a specific ISD project, we obtained a set of propositions concerning the language development process during requirements development that we derived from our interpretation and analysis of the data (cf. Table 5). This gives evidence for the practicability and utility of our procedure and our concepts.

The application of the concept of language quality provides a lens for researchers who are looking for a suitable means to analyze the emergence of consensus and shared understanding in ISD processes. At the same time, we provide practitioners with first insights and create awareness for simple, yet possible deficiencies and pitfalls that have a real impact on ISD project performance. For example, we specifically conclude from our findings that to achieve shared understanding it might be very useful to strive for an explicit agreement on representing symbols for relevant concepts in order to avoid the emergence of volatile symbols, long discussions, or uncertainty (P1, P2, P3). Our findings indicate that explicitness in the achievement of language quality does not occur in a self-evident way (P4). Furthermore, our data indicate that, because of the high frequency of appearance of deficiencies, the impact of using volatile symbols on language quality in the long run is a question worth addressing in the future (P5). Finally, we present evidence why language quality is definitely important to the field of ISD, especially in requirements elicitation and negotiation (P6).

However, the analysis of language quality during language development in ISD is challenging for different reasons. It is not only the volatile nature of oral communication that makes dealing with it difficult. More important is the fact that language use is an attempt to bring different kinds of individual, domain-specific knowledge to the surface of social interaction. For example, the use of narrative elements in examples of other ISD cases suggests that a low language quality is inherent to early stages of the ISD process (Alvarez and Urla, 2002). Although we do not assert that it is possible to avoid low language quality during early stages of language development, we argue that there are important concepts and definitions in every ISD process for which a low language quality, in the long run, is unnecessary and disadvantageous. This also suggests that some of the reasons why agile methodologies are not a 'silver bullet' (Berry, 2004) in ISD for every problem and situation at hand are related to the practicalities of informal communication and language. On the one hand, agile development teams continuously gather and adjust requirements, with all the stakeholders (for example, customers, users, developers) participating actively and communicating intensively with each other at all times; thereby they constantly tune and change their understanding of requirements, adjust the used language, and may well detect issues of language quality earlier than traditional, more formal development teams. On the other hand, purely informal and oral communication might lead to more situation- or context-specific symbols and low language quality, which might cause problems in situations with many stakeholders that have to build shared understanding, all having diverse backgrounds and context- or domain-specific knowledge (Cao *et al*., 2009: 333), or when ISD knowledge is needed later for managing long-running systems and needs to be formalized and codified (Black *et al*., 2009: 39). More studies are needed of how different development methodologies affect language quality in different situations and settings with varying contextual factors.

The kind of analysis such as the one presented in this article also may be helpful for uncovering problems in global, virtual teams in offshoring projects considering the question how symbols and concepts between different native languages are linked. In fact, although the used natural language in the project we observed was German, the example of OD-19 that we presented shows that the stakeholders used the English words 'case study', its German translation 'Fallstudie', and the German translation 'Erfahrungsbericht' (a part-different translation meaning 'field report') to address the same concept. The dimension of redundancy helps us to make this source of misunderstandings clear, and it also enables detailed insights into the wide spectrum of possible reactions of stakeholders during social interactions. What do stakeholders do in such cases? Do they communicate about this redundancy? Do they try to solve redundancy? Do they make use of it to make sure they understand the meaning? In the end, requirements need to be understood in the same way – whatever the conscious motivation or unconscious reason stakeholders' may have to use redundancy. For addressing multilingual offshoring projects, it may be sufficient to agree on the definition of, for example, one concept and one symbol for each native language as the ideal of language quality.

Our work is not without limitations. Our conceptualization of language quality focuses only on the emergence of shared understanding and ignores other aspects of language in relation to social interactions. We acknowledge that other aspects of the dynamics of social interaction are important and that several perspectives exist for examining social interactions. In terms of Giddens' (1984) dimensions of social structure, we solely deal with the dimension of *signification* (including communication and interpretative schemes) and exclude – for concerns of research focus and analyzability – other dimensions such as domination or legitimation. Furthermore, language is not the only way to look at social interactions. Other factors and conditions may also have key roles in requirements development, for example, stakeholders' skills, capabilities, characteristics and goals, their institutional contexts, power, culture, or norms (for example, Robey and Markus, 1984: Iivari and Iivari, 2011). It is also important to note that our analysis focused only on the *observable level* of communication. We are only able to deduce different concepts by interpretation of these observations and their context. Consequently, we are not targeting deficiencies that can only be uncovered by looking into people's minds, but deficiencies whose sources are processes visible to everyone, yet not taken into consideration. We acknowledge other areas dealing with the dynamics of social communication, such as the analysis of gestures, problems of culture, or issues of social power as important (Rowlands, 2009). Pragmatically, there may be several reasons why language quality may be disrupted. In our research, we focus only on causes that emerge from the pragmatic level in oral communication and language use.

We assume that in our observed field of requirements development during ISD, language quality has relevance, which needs to be assessed. With this research, we hope to provide a useful approach for analysis while at the same time contributing to practitioners' views of the language development process during ISD. In general, we assert that the lens of language quality can provide a useful approach for analyzing and understanding the processes of language development. This seems to be a suitable method for the analysis of communication using natural language, for example, in comparing projects with formal mechanisms of communication such as documents or models to projects using informal mechanisms such as face-to-face communication. For qualitative IS researchers, this is very useful because our approach can complement the usual interview-based studies by providing a guide for 'contextualized accounts of meanings' (Silverman, 1998: 19).

For practitioners, it will be helpful to know the consequences of different actions on language quality, which certainly requires a deeper analysis of them – something that we were able to present during this research. An analysis of language quality could thus be used to complement approaches such as storytelling or the persona-scenario method (Madsen and Nielsen, 2010). For example, phases where system analysts assess the language quality of their requirements documents or their stakeholder interviews could be applied as a pre-check within current requirements development methodologies (for example, Dietz and Juhrisch, 2012). With recent advances in algorithms for text mining and analysis (for example, Landauer *et al*., 2007; Evangelopoulos *et al*., 2012; Holten *et al*., 2010), it may even become possible to provide limited computer-based support by parsing snippets of conversations, detecting possible deficiencies in language quality, and signalling those possible deficiencies to stakeholders to raise their awareness – although we think that this task cannot be fully automated (cf. section 'Information Systems Development and Requirements Development as a Language Development Process'). In addition, our research may offer a foundation for the design of evaluation and training tools dealing with the quality of social interactions of ISD stakeholders. Our approach offers (a) a very clear notion of the reasons behind the problem of low language quality and (b) a systematic way to analyze and trace the problem's emergence. For example, an introductory session on 'language quality' could be a topic for a training unit for ISD practitioners, including short interaction experiments. Afterwards, the attention of training participants could be drawn on the interaction course in their experiment, on causes for bottlenecks and breakdowns, and on their own role concerning the improvement of the overall language quality. We suggest that a sensitization of all project stakeholders about possible sources of deficiencies in language quality can bring positive results for the outcome of the ISD project. This should benefit the whole project team by helping them to build a better shared understanding of requirements more quickly. We fully embrace the position of Curtis *et al.* (1988): 'When we overlay cognitive, social, and organizational processes on the phased evolution of software artifacts, we begin to see causes for bottlenecks and inefficiencies in development. The more deeply project managers understand these behavioural processes, the greater their insight into the factors that determine their success' (p. 1284). Future studies can build on our insights and, in the end, work on methods, tools, and approaches that support ISD.

Conclusion

In this article, we have proposed an approach to analyzing language use and communication in requirements development during ISD in a unique and novel way. We emphasize that language is an important 'variable of interest' (Pondy, 2005: 133) and should become a central feature of study in ISD. We suggest that our concept of language quality illuminates some of the factors that lead to specific patterns and phenomena in ISD. If our concepts find acceptance in the ISD research community, they can be used to study ISD phenomena and to better understanding the ISD process. Although the main contribution of the article is the new research perspective on social interactions during ISD as a language development process, we also formulate implications of our findings for ISD practitioners and link our findings back to the discussion of 'agility *vs* formality' in ISD studies. We hope that our concepts will be useful for other ISD researchers and practitioners.

Our next steps will be to gather more empirical data in order to corroborate our propositions. Further research that builds on broader data will also be able to address the ability of different stakeholder groups to achieve and retain a shared understanding. We also plan to investigate the role of documents, print outs, data and process conceptual models, or program code in verbal communication, which are instances of the 'design boundary objects' (Bergman *et al*., 2007) discussed in the ISD literature. Furthermore, a comparison between different degrees of expertise or domain-specific knowledge in ISD may bring to light more suitable strategies for successfully addressing misunderstandings and unnecessary low language quality. Finally, social interactions in ISD are not confined to the interaction taking place within the project team. An investigation of language quality in communication between project teams and other stakeholders such as business managers or with former team members who have left the group at one point and are returning to discuss progress with their group, whom in a sense they represent, could be subject to more or other language quality issues.

Acknowledgements

We would like to thank the editorial team of *Journal of Information Technology* for guidance and the three anonymous reviewers for very supportive feedback that helped enormously to improve the manuscript. The conference audience at ICIS 2010 also contributed with valuable comments to an earlier version of this article. In addition we would like to thank Christian Bill for his help in analyzing the data and Dr. Karlheinz Kautz for fruitful discussions and comments. Parts of this study were funded by the German Federal Ministry of Education (record no.: 01HQ0608).

Note

1 The same data set was used to develop the underlying framework in a companion paper (Corvera Charaf *et al*., 2012), although with a focus on the *process* of language development, whereas this article has a focus on the quality of the shared language that is built as the *product* of this process.

References

Abran, A., Moore, J.W., Bourque, P. and Dupuis, R. (eds.) (2004). *Guide to the Software Engineering Body of Knowledge (SWEBOK®)*, Los Alamitos, CA: IEEE Computer Society.

Agrawal, M. and Chari, K. (2007). Software Effort, Quality, and Cycle Time: A study of CMM Level 5 projects, *IEEE Transactions on Software Engineering* **33**(3): 145–156.

Alvarez, R. and Urla, J. (2002). Tell Me a Good Story: Using narrative analysis to examine information requirements interviews during an ERP implementation, *The DATA BASE for Advances in Information Systems* **33**(1): 38–52.

Alvesson, M. (2003). Methodology for Close up Studies – Struggling with closeness and closure, *Higher Education* **46**(2): 167–193.

Ambriola, V. and Gervasi, V. (1997). Processing Natural Language Requirements, *12th IEEE International Conference on Automated Software Engineering (ASE'97) (formerly: KBSE)*, Lake Tahoe, USA, p. 36.

Ambriola, V. and Gervasi, V. (2006). On the Systematic Analysis of Natural Language Requirements with CIRCE, *Automated Software Engineering* **13**(1): 107–167.

Anupindi, R., Chopra, S., Deshmukh, S.D., Van Mieghem, J.A. and Zemel, E. (2006). *Managing Business Process Flows. Principles of Operations Management*, 2nd edn, Upper Saddle River, NJ, USA: Pearson Education.

Auramäki, E., Lehtinen, E. and Lyytinen, K. (1988). A Speech-act-based Office Modeling Approach, *ACM Transactions on Information Systems (TOIS)* **6**(2): 126–152.

Austin, J.L. (1962). *How to do Things with Words: The William James Lectures delivered at Harvard University in 1955*. in J.O. Urmson (ed.), Oxford, UK: Clarendon.

Avison, D.E. and Fitzgerald, G. (2003). Where Now for Development Methodologies? *Communications of the ACM* **46**(1): 78–82.

Beck, K. and Andres, C. (2004). *Extreme Programming Explained: Embrace Change*, 2nd edn, Boston, MA, USA, Addison-Wesley Professional.

Bergman, M., Lyytinen, K. and Mark, G. (2007). Boundary Objects in Design: An ecological view of design artifacts, *Journal of the Association for Information Systems* **8**(11): 546–568.

Berry, D.M. (2004). The Inevitable Pain of Software Development: Why there is no Silver Bullet, in M. Wirsing, A. Knapp and S. Balsamo (eds.) *Radical Innovations of Software and Systems Engineering in the Future*, Berlin, Heidelberg, Germany: Springer, pp. 50–74.

Berry, D.M. and Kamsties, E. (2003). Ambiguity in Requirements Specification, in S.d.P. Leite and K.J.H. Doorn (eds.) *Perspectives on Software Requirements*, Boston, Dordrecht, London: Kluwer, pp. 191–194.

Beynon-Davies, P. (2010). The Enactment of Significance: A unified conception of information, systems and technology, *European Journal of Information Systems* **19**(4): 389–408.

Black, S.E., Boca, P.P., Bowen, J.P., Gorman, J. and Hinchey, M.G. (2009). Formal versus Agile: Survival of the fittest, *IEEE Computer* **49**(9): 39–45.

Blair, D.C. (2005). Wittgenstein, Language and Information: "Back to the Rough Ground!", in F. Crestani and I. Ruthven (eds.) *5th International Conference on Conceptions of Library and Information Sciences (CoLIS 2005)*, Glasgow, UK: Springer, p. 1.

Boehm, B. and Basili, V.R. (2000). Gaining Intellectual Control of Software Development, *Computer* **33**(5): 27–33.

Boehm, B. and Turner, R. (2004). *Balancing Agility and Discipline: A guide for the perplexed*, Boston, MA, USA: Addison-Wesley.

Boehm, B.W. (1988). A Spiral Model of Software Development and Enhancement, *IEEE Computer* **21**(4): 61–72.

Boh, W.F., Slaughter, S.A. and Espinosa, J.A. (2007). Learning from Experience in Software Development: A multilevel analysis, *Management Science* **53**(8), August 1, 2007, 1315–1331.

Boland, R.J.J. (1979). Control, Causality, and Information System Requirements, *Accounting, Organization and Society* **4**(4): 259–272.

Boroditsky, L., Fuhrman, O. and McCormick, K. (2010). Do English and Mandarin Speakers Think Differently about Time? *Cognition* **118**(1): 123–129.

Boroditsky, L. and Gaby, A. (2010). Remembrances of Times Eastt: Absolute spatial representations of time in an Australian Aboriginal community, *Psychological Science* **21**(11): 1635–1639.

Bostrom, R. and Heinen, J.S. (1977). MIS Problems and Failures: A socio-technical perspective, *MIS Quarterly* **1**(3): 11–28.

Bostrom, R.P. (1989). Successful application of communication techniques to improve the systems development process, *Information & Management* **16**(5): 279–295.

Branigan, H.P., Pickering, M.J., Pearson, J. and McLean, J.F. (2010). Linguistic Alignment between People and Computers, *Journal of Pragmatics* **42**(9): 2355–2368.

Brighton, H. and Kirby, S. (2001). The Survival of the Smallest: Stability conditions for the cultural evolution of compositional language, in J. Kelemen and P. Sosík (eds.) *Advances in Artificial Life*, Berlin, Germany: Springer, pp. 592–601.

Byrd, T.A., Cossick, K.L. and Zmud, R.W. (1992). A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques, *MIS Quarterly* **16**(1): 117–138.

Cao, L., Mohan, K., Peng, X. and Ramesh, B. (2009). A Framework for Adapting Agile Development Methodologies, *European Journal of Information Systems* **18**(4): 332–343.

Carnap, R. (1956). *Meaning and Necessity: A Study in semantics and modal logic*, 2nd edn, Chicago, IL, USA: The University of Chicago Press.

Chae, B. and Poole, M.S. (2005). The Surface of Emergence in Systems Development: Agency, institutions, and large-scale information systems, *European Journal of Information Systems* **14**(1): 19–36.

Chakraborty, S., Sarker, S. and Sarker, S. (2010). An Exploration into the Process of Requirements Elicitation: A grounded approach, *Journal of the Association for Information Systems* **11**(4): 212–249.

Clark, H.H. (1996). *Using Language*, New York: Cambridge University Press.

Clarke, R.J. (2001). Studies in Organisational Semiotics: An introduction, *Australasian Journal of Information Systems* **8**(2): 75–82.

Cohn, M.L., Sim, S.E. and Lee, C.P. (2009). What Counts as Software Process? Negotiating the boundary of software work through artifacts and conversation, *Computer Supported Cooperative Work* **18**(5–6): 401–443.

Corvera Charaf, M., Rosenkranz, C. and Holten, R. (2012). The Emergence of Shared Understanding: Applying functional pragmatics to study the requirements development process, *Information Systems Journal*, doi: 10.1111/j.1365-2575.2012.00408.x.

Cule, P., Schmidt, R., Lyytinen, K. and Keil, M. (2000). Strategies for Heading off IS Project Failure, *Information Systems Management* **17**(2): 65–73.

Curtis, B., Krasner, H. and Iscoe, N. (1988). A Field Study of the Software Design Process for Large Systems, *Communications of the ACM* **31**(11): 1268–1287.

Davidson, E.J. (2002). Technology Frames and Framing: A socio-cognitive investigation of requirements determination, *MIS Quarterly* **26**(4): 329–358.

Davis, A.M. (1990). The Analysis and Specification of Systems and Software Requirements, in R.H. Thayer and M. Dorfman (eds.) *Systems and Software Requirements Engineering*, Los Alamitos, CA, USA: IEEE Computer Society Press-Tutorial, pp. 119–134.

Deacon, T.W. (1997). *The Symbolic Species. The co-evolution of language and the brain*, New York: W. W. Norton & Company.

Deacon, T.W. (2005). Language as an Emergent Function: Some radical neurological and evolutionary implications, *Theoria* **54**(20): 269–286.

DeMarco, T. and Lister, T. (1987). *Peopleware: Productive projects and teams*, New York: Dorset House Publishing Co., Inc.

de Saussure, F. (1974). *Course in General Linguistics*, London: Peter Owen Ltd.

Dietz, G. and Juhrisch, M. (2012). Negotiating Language Barriers – A Methodology for Cross-Organisational Conceptual Modelling[dagger], *European Journal of Information Systems* **21**(3): 229–254.

Dwivedi, Y.K., Lal, B. and Williams, M.D. (2009). *Handbook Of Research On Contemporary Theoretical Models In Information Systems*, Hershey, PA, USA: IGI Global.

Dybå, T. and Dingsøyr, T. (2009). What do we know about agile software development? *IEEE Software* **26**(5): 6–9.

Espinosa, J.A., Slaughter, S.A., Kraut, R.E. and Herbsleb, J.D. (2007). Team Knowledge and Coordination in Geographically Distributed Software Development, *Journal of Management Information Systems* **24**(1): 135–169.

Evangelopoulos, N., Zhang, X. and Prybutok, V.R. (2012). Latent Semantic Analysis: Five methodological recommendations, *European Journal of Information Systems* **21**(1): 70–86.

Falkenberg, E.D., Hesse, W., Lindgreen, P., Nilsson, B.E., Oei, J.L.H., Rolland, C., Stamper, R.K., Assche, F.J.M.V., Verrijn-Stuart, A.A. and Voss, K. (1998). A Framework of Information Systems Concepts, The FRISCO Report, IFIP WG 8.1 Task Group FRISCO (Web edition). [WWW document] http://www.mathematik.uni-marburg.de/~hesse/papers/fri-full.pdf (accessed 23 September 2009).

Fausey, C. and Boroditsky, L. (2011). Who Dunnit? Cross-Linguistic differences in Eye-Witness Memory, *Psychonomic Bulletin & Review* **18**(1): 150–157.

Fichman, R.G. and Kemerer, C.F. (1993). Adoption of Software Engineering Process Innovations: The case of object orientation, *Sloan Management Review* **34**(2): 7–22.

Fitzgerald, B., Hartnett, G. and Conboy, K. (2006). Customising Agile Methods to Software Practices at Intel Shannon, *European Journal of Information Systems* **15**(2): 200–213.

Flores, F., Graves, M., Hartfield, B. and Winograd, T. (1988). Computer Systems and the Design of Organizational Interaction, *ACM Transactions on Information Systems* **6**(2): 153–172.

Fraser, S. and Mancl, D. (2008). No Silver Bullet: Software engineering reloaded, *IEEE Software* **25**(1): 91–94.

Galliers, R.D. and Swan, J.A. (2000). There's More to Information Systems Development than Structured Approaches: Information requirements analysis as a socially mediated process, *Requirements Engineering* **5**(2): 74–82.

Gallivan, M.J. and Keil, M. (2003). The User-Developer Communication Process: A critical case study, *Information Systems Journal* **13**(1): 37–68.

Gasson, S. (1999). A Social Action Model of Situated Information Systems Design, *The DATA BASE for Advances in Information Systems* **30**(2): 82–97.

Gervasi, V. and Zowghi, D. (2005). Reasoning About Inconsistencies in Natural Language Requirements, *ACM Transactions on Software Engineering and Methodology* **14**(3): 277–330.

Giddens, A. (1984). *The Constitution of Society: Outline of theory of structuration*, Berkley: University of California Press.

Goeken, M. and Patas, J. (2010). Evidenzbasierte Strukturierung und Bewertung empirischer Forschung im Requirements Engineering, *WIRTSCHAFTSINFORMATIK* **52**(3): 173–184.

Goldkuhl, G. and Lyytinen, K. (1982). A language action view of information systems, *3rd International Conference on Information Systems (ICIS 1982)*, Houston, TX, USA.

Gopal, A. and Prasad, P. (2000). Understanding GDSS in Symbolic Context: Shifting the focus from technology to interaction, *MIS Quarterly* **24**(3): 509–546.

Guinan, P. and Bostrom, R.P. (1986). Development of Computer-based Information Systems: A communication framework, *SIGMIS Database* **17**(3): 3–16.

Guinan, P.J. and Scudder, J.N. (1989). Client-Oriented Interactional Behaviors for Professional-Client Settings, *Human Communication Research* **15**(3): 444–462.

Hansen, S., Berente, N. and Lyytinen, K.J. (2008). Requirements in the 21st Century: Current practice & emerging trends, in K.J. Lyytinen, P. Loucopoulos, J. Mylopoulos and W. Robinson (eds.) *Design Requirements Engineering: A Ten-Year Perspective*, Berlin, Germany: Springer, pp. 44–87.

Hansen, S. and Lyytinen, K. (2009). Distributed Cognition in the Management of Design Requirements, in M. Jarke, K. Lyytinen and J. Mylopoulos (eds.) *Perspectives Workshop: Science of Design: High-Impact requirements for software-intensive systems*, Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Germany.

Hansen, S. and Lyytinen, K. (2010). Challenges in Contemporary Requirements Practice, *43th Hawaii International Conference on System Sciences (HICSS 2010)*, Koloa, HI, USA.

Hansen, S. and Rennecker, J. (2006). Collective Hermeneutics in a Systems Development Process, in *Sprouts: Working Papers on Information Systems*.

Hansen, S. and Rennecker, J. (2010). Getting on the Same Page: Collective hermeneutics in a systems development team, *Information and Organization* **20**(1): 44–63.

He, J., Butler, B.S. and King, W.R. (2007). Team Cognition: Development and evolution in software project teams, *Journal of Management Information Systems* **24**(2): 261–292.

Herbsleb, D. and Mockus, A. (2003). An Empirical Study of Speed and Communication in Globally-Distributed Software Development, *IEEE Transactions on Software Engineering* **29**(6): 1–14.

Hesse, W., Müller, D. and Ruß, A. (2008). Information, Information Systems, Information Society: Interpretations and implications, *Poiesis & Praxis: International Journal of Technology Assessment and Ethics of Science* **5**(3): 159–183.

Hirschheim, R., Klein, H. and Lyytinen, K. (1995). *Information Systems Development and Data Modeling. Conceptual and philosophical foundations*, Cambridge, UK: Cambridge University Press.

Holmqvist, B. (1989). Work, Language and Perspective: An empirical investigation of the interpretation of computer-based information systems, *Scandinavian Journal of Information Systems* **1**(1): 72–96.

Holten, R. and Rosenkranz, C. (2011). Designing Viable Social Systems: The role of linguistic communication for self-organization, *Kybernetes* **40**(3/4): 559–580.

Holten, R., Rosenkranz, C. and Kolbe, H. (2010). Measuring Application Domain Knowledge: Results from a preliminary experiment, *31st International Conference on Information Systems (ICIS 2010)*, St. Louis, Missouri, USA.

Hoppenbrouwers, S. and Weigand, H. (2000). Meta-Communication in the Language Action Perspective, in M. Schoop and C. Quix (eds.) *Proceedings of the Fifth International Workshop on the Language-Action Perspective on Communication Modelling (LAP 2000)*, Aachen, Germany, University of Tilburg.

Iivari, J., Hirschheim, R. and Klein, H.K. (2004). Towards a Distinctive Body of Knowledge for Information Systems experts: Coding ISD process knowledge in two IS journals, *Information Systems Journal* **14**(4): 313–342.

Iivari, J. and Iivari, N. (2011). The Relationship between Organizational Culture and the Deployment of Agile Methods, *Information and Software Technology* **53**(5): 509–520.

Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J. and Robinson, W. (2010). The Brave New World of Design Requirements: Four key principles, in B. Pernici (ed.) *Advanced Information Systems Engineering*, Berlin/Heidelberg: Springer, pp. 470–482.

Jarke, M. and Lyytinen, K. (2010). High Impact Requirements Engineering, *WIRTSCHAFTSINFORMATIK* **52**(3): 115–116.

Jarke, M., Lyytinen, K., Mylopoulos, J., Kappel, G., Leite, J., Mark, G., Ramesh, B., Schmitz, D. and Sutcliffe, A.G. (2009). High-Impact Requirements for Software-Intensive Systems: Seminar Outlines and Working Group Summaries, in M. Jarke, K. Lyytinen and J. Mylopoulos (eds.) *Perspectives Workshop: Science of Design: High-Impact requirements for software-intensive systems*, Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Germany.

Jarke, M. and Pohl, K. (1993). Establishing Visions in Context: Towards a model of requirements processes, *Proceedings 14th International Conference on Information Systems*, Orlando, FL, USA, 23–34.

Joshi, K.D., Sarker, S. and Sarker, S. (2007). Knowledge Transfer within Information Systems development Teams: Examining the role of knowledge source attributes, *Decision Support Systems* **43**(2): 322–335.

Kaasbøll, J. (1987). Intentional Development of Professional Language through Computerization. A case study and some theoretical considerations, in P. Docherty, K. Fuchs-Kittowski, P. Kolm and L. Mathiassen (eds.) *Proceeding of the IFIP TC 9/WG 9.1 Working Conference on System Design for Human Development and Productivity: Participation and beyond*, Berlin, Germany: Elsevier, 371–382.

Kamlah, W. and Lorenzen, P. (1984). *Logical Propaedeutic. Pre-School of reasonable discourse*, Lanham: University Press of America.

Kautz, K., Madsen, S. and Nørbjerg, J. (2007). Persistent Problems and Practices in Information Systems Development, *Information Systems Journal* **17**(3): 217–239.

Kautz, K. and Nielsen, P.A. (2004). Understanding the Implementation of Software Process Improvement Innovations in Software Organizations, *Information Systems Journal* **14**(1): 3–22.

Ko, D.-G., Kirsch, L.J. and King, W.R. (2005). Antecedents of Knowledge Transfer from Consultants to Clients in Enterprise System Implementations, *MIS Quarterly* **29**(1): 59–85.

Kraut, R.E. and Streeter, L.A. (1995). Coordination in Software Development, *Communications of the ACM* **38**(3): 69–81.

Land, F. and Somogyi, E. (1986). Software Engineering: The relationship between a formal system and its environment, *Journal of Information Technology (Routledge, Ltd.)* **1**(1): 02, 14.

Landauer, T.K., McNamara, D.S., Dennis, S. and Kintsch, W. (eds.) (2007). *Handbook of Latent Semantic Analysis*, Mahwah: Lawrence Erlbaum Associates.

Landis, J.R. and Koch, G.G. (1977). The Measurement of Observer Agreement for Categorical Data, *Biometrics* **33**(1): 159–174.

Lee, A.S. (1991). Integrating Positivist and Interpretive Approaches to Organizational Research, *Organization Science* **2**(4): 342–365.

Lee, G. and Xia, W. (2010). Toward Agile: An integrated analysis of quantitative and qualitative field data, *MIS Quarterly* **34**(1): 87–114.

Levina, N. (2005). Collaborating on Multiparty Information Systems Development Projects: A collective reflection-in-action view, *Information Systems Research* **16**(2): 109–130.

Levina, N. and 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.

Loucopoulos, P. and Karakostas, V. (1995). *System Requirements Engineering*, Maidenhead, Berkshire: McGraw-Hill International Series in Software Engineering.

Lyytinen, K. and Newman, M. (2008). Explaining Information Systems change: A punctuated socio-technical change model, *European Journal of Information Systems* **17**(6): 589–613.

Lyytinen, K. and Robey, D. (1999). Learning Failure in Information Systems Development, *Information Systems Journal* **9**(2): 85–101.

Lyytinen, K.J. (1985). Implications of Theories of Language for Information Systems, *MIS Quarterly* **9**(1): 61–76.

Madsen, S. and Nielsen, L. (2010). Exploring Persona-Scenarios – Using storytelling to create design ideas, in D. Katre, R. Orngreen, P. Yammiyavar and T. Clemmensen (eds.) *Human Work Interaction Design: Usability in Social, Cultural and Organizational Contexts*, Boston: Springer, pp. 57–66.

Marakas, G.M. and Elam, J.J. (1998). Semantic Structuring in Analyst Acquisition and Representation of facts in Requirements Analysis, *Information Systems Research* **9**(1): 37–63.

Maryam, A. (1984). An Assessment of the Prototyping Approach to Information Systems Development, *Communications of the ACM* **27**(6): 556–563.

Mathiassen, L., Tuunanen, T., Saarinen, T. and Rossi, M. (2007). A Contingency Model for Requirements Development, *Journal of the Assoication for Information Systems* **8**(11): 569–597.

McMaster, M. and Grinder, J. (1980). *Precision: A New Approach to Communication*, Beverly Hills, CA: Precision Models.

Milroy, L. (1984). Comprehension and Context: Successful communication and communicative breakdown, in P. Trudgill (ed.), *Applied Sociolinguistics*, London: Academic, pp. 7–31.

Moldoveanu, M.C. (2002). Language, Games and Language Games, *Journal of Socio-Economics* **31**(3): 233–251.

Morrison, J. and George, J.F. (1995). Exploring the Software Engineering Component in MIS Research, *Communications of the ACM* **38**(7): 80–91.

Nahapiet, J. and Ghoshal, S. (1998). Social Capital, Intellectual Capital, and the Organizational Advantage, *The Academy of Management Review* **23**(2): 242–266.

Nelson, R.R. (2007). IT Project Management: Infamous failures, classic mistakes, and best practices, *MIS Quarterly Executive* **6**(2): 67–78.

Newman, M. and Robey, D. (1992). A Social Process Model of User-Analyst Relationships, *MIS Quarterly* **16**(2): 249–266.

Ogden, C.K. and Richards, I.A. (1923). *The Meaning of Meaning: A study of the influence of language upon thought and of the science of symbolism*, London: Routledge & Kegan Paul.

Orlikowski, W.J. and Gash, D.C. (1994). Technological Frames: Making sense of information technology in organizations, *ACM Transactions on Information Systems* **12**(2): 174–207.

Orlikowski, W.J. and Robey, D. (1991). Information Technology and the Structuring of Organizations, *Information Systems Research* **2**(2): 143–169.

Pask, G. (1975). *Conversation, Cognition and Learning: A cybernetic theory*, New York: Elsevier.

Pernille, B. and Ojelanki, N. (2009). Virtual Team Collaboration: Building shared meaning, resolving breakdowns and creating translucence, *Information Systems Journal* **19**(3): 227–253.

Pickering, M.J. and Garrod, S. (2004). Toward a Mechanistic Psychology of Dialogue, *Behavioral and Brain Sciences* **27**(2): 169–226.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P. and Still, J. (2008). The Impact of Agile Practices on Communication in Software Development, *Empirical Software Engineering* **13**(3): 303–337.

Pohl, K. (1993). The Three Dimensions of Requirements Engineering, *5th International Conference of Advanced Information Systems Engineering (CAISE'93)*, Paris, France: Springer Verlag, Berlin, 275–292.

Pohl, K. (1994). The Three Dimensions of Requirements Engineering: A Framework and its Applications, *Information Systems* **19**(3): 243–258.

Pondy, L.R. (2005). Beyond Open System Models of Organization, Reprinted in E:CO. (7:3–4), pp. 119–137. Originally presented at the Annual Meeting of the Academy of Management, Kansas City, Missouri, August 112, 1976.

Poppendieck, M. and Poppendieck, T. (2003). *Lean Software Development: An Agile toolkit*, 1st edn, Amsterdam: Addison-Wesley Longman.

Ramesh, B., Cao, L. and Baskerville, R. (2010). Agile Requirements Engineering Practices and Challenges: An empirical study, *Information Systems Journal* **20**(5): 449–480.

Robey, D. and Markus, M.L. (1984). Rituals in Information System Design, *MIS Quarterly* **8**(1): 5–15.

Robey, D. and Newman, M. (1996). Sequential Patterns in Information Systems Development: An application of a social process model, *ACM Transactions on Information Systems (TOIS)* **14**(1): 30–63.

Robillard, P.N. (1999). The Role of Knowledge in Software Development, *Communications of the ACM* **42**(1): 87–92.

Rowlands, B. (2009). A Social Actor Understanding of the Institutional Structures at Play in Information Systems Development, *Information Technology & People* **22**(1): 51–62.

Royce, W.W. (1970). Managing the Development of Large Software Systems: Concepts and techniques, Proceedings of WesCon, Reprint in: Proceedings of the 9th ICSE 1987: ICSE, 87 Proceedings of the 9th International Conference on Software Engineering, Los Alamitos, CA: IEEE Computer Society Press, 328–338.

Sambamurthy, V. and Kirsch, L.J. (2000). An Integrative Framework of the Information Systems Development Process, *Decision Sciences* **31**(2): 391–411.

Sawyer, S. and Guinan, P.J. (1998). Software Development: Processes and performance, *IBM Systems Journal* **37**(4): 552–569.

Sawyer, S., Guinan, P.J. and Cooprider, J. (2010). Social Interactions of Information Systems development Teams: A performance perspective, *Information Systems Journal* **20**(1): 81–107.

Schoop, M. (2001). An Introduction to the Language-Action Perspective, *CM SIGGROUP Bulletin* **22**(2): 3–8.

Schutz, A. (ed.) (1962). Concept and Theory Formation in the Social Sciences, *Collected Papers: The problem of social reality*, The Hague: Martinus Nijhoff, pp. 48–66.

Schwaber, K. (1995). Scrum Development Process, *Conference on Object-Oriented Programming Systems, Languages, and Applications*, pp. 117–134.

Searle, J.R. (1969). *Speech Acts: An essay in the philosophy of language*, Cambridge, UK: University Press.

Selten, R. and Warglien, M. (2007). The Emergence of Simple Language in an Experimental Coordination Game, *Proceedings of the National Academy of Sciences* **104**(18): 7361–7366.

Shenhav, Y., Alon, S. and Shrum, W. (1994). Goodness' Concepts in the Study of Organizations: A longitudinal survey of four leading journals, *Organization Studies* **15**(5), 09/01, 753–776.

Siau, K., Long, Y. and Ling, M. (2010). Toward a Unified Model of Information Systems Development Success, *Journal of Database Management* **21**(1): 80–101.

Silverman, D. (1998). Qualitative Research: Meanings or practices, *Information Systems Journal* **8**(1): 3–20.

Smith, K. (2004). The Evolution of Vocabulary, *Journal of Theoretical Biology* **228**(1): 127–142.

Sommerville, I. (2001). *Software Engineering*, 6th edn, Essex: Pearson Education Limited.

Stamper, R. (1973). *Information in Business and Administrative Systems*, New York: John Wiley & Sons.

Stamper, R., Liu, K., Hafkamp, M. and Ades, Y. (2000). Understanding the Roles of Signs and Norms in Organizations – A Semiotic Approach to Information Systems Design, *Behaviour & Information Technology* **19**(1): 15–27.

Tan, M. (1994). Establishing Mutual Understanding in Systems Design: An Empirical Study, *Journal of Management Information Systems* **10**(4): 159–182.

Tomasello, M. (2008). *Origins of Human Communication*, Cambridge, MA: MIT Press.

Tomasello, M., Carpenter, M., Call, J., Behne, T. and Moll, H. (2005). Understanding and Sharing Intentions: The origins of cultural cognition, *Behavioral and Brain Sciences* **28**(5): 675–691.

Truex, D.P., Baskerville, R. and Klein, H. (1999). Growing Systems in Emergent Organizations, *Communications of the ACM* **42**(8): 117–123.

Tsoukas, H. (2005). Afterword: Why language matters for the analysis of organizational change, *Journal of Organizational Change Management* **18**(1): 96–104.

Urquhart, C. (2001). Analysts and Clients in Organisational Contexts: A conversational perspective, *The Journal of Strategic Information Systems* **10**(3): 243–262.

Vidgen, R. and Wang, X. (2009). Coevolving Systems and the Organization of Agile Software Development, *Information Systems Research* **20**(3): 355–376.

Wand, Y. and Wang, R. (1996). Anchoring Data Quality Dimensions in Ontological Foundations, *Communications of the ACM* **39**(11): 86–95.

Wand, Y. and Weber, R. (1995). On the Deep Structure of Information Systems, *Information Systems Journal* **5**(3): 203–223.

Watts Sussman, S. and Guinan, P.J. (1999). Antidotes for High Complexity and Ambiguity in Software Development, *Information & Management* **36**(1): 23–35.

Weber, R.A. and Camerer, C.F. (2003). Cultural Conflict and Merger Failure: An experimental approach, *Management Science* **49**(4): 400–415.

Williams, C. (2010). Client–Vendor Knowledge Transfer in IS Offshore Outsourcing: Insights from a survey of Indian software engineers, *Information Systems Journal* **21**(4): 335–356.

Winawer, J., Witthoft, N., Frank, M.C., Wade, A.R., Wu, L. and Boroditsky, L. (2007). Russian Blues Reveal Effects of Language on Color Discrimination, *Proceedings of the National Academy of Sciences* **104**(19): 7780–7785.

Winograd, T. (1988). A Language/Action Perspective on the Design of Cooperative Work, *Human-Computer Interaction* **3**(1): 3–30.

Winograd, T. and Flores, F. (1986). *Understanding Computers and Cognition: A new foundation for design*, Norwood: Ablex Publishing Corp.

Wynn, E.H., Whitley, E.A. and Myers, M. (2002). Placing Language in the Foreground: Themes and methods in information technology discourse, in E.H. Wynn, E.A. Whitley, M.D. Myers and J.I.D. Gross (eds.) *Global and Organizational Discourse about Information Technology*, Boston: Kluwer Academic Publishers, pp. 1–12.

Xia, W. and Lee, G. (2005). Complexity of Information Systems Development Projects: Conceptualization and measurement development, *Journal of Management Information Systems* **22**(1): 45–83.

About the authors

Christoph Rosenkranz is an Assistant Professor (Akademischer Rat auf Zeit) at the Department of Economics and Business Administration, Goethe-University Frankfurt, Germany, where he also received his Ph.D. in Information Systems. His research focuses on information systems development and requirements engineering, business intelligence and business process management, organization theory, and online communities.

Marianne Corvera Charaf is a consultant and researcher specializing in business strategy, process analysis and improvement, data quality, and information systems development. She obtained her Ph.D. in Information Systems at the Department of Economics and Business Administration, Goethe-University Frankfurt.

Roland Holten is a Full Professor of Information Systems and holds the Chair for Information Systems Engineering at Goethe-University Frankfurt, Germany. He received his Ph.D. in Information Systems from the Westfälische Wilhelms-Universität Münster, Germany. His research interests include IT-mediated communication structures of groups, information systems development, and IS modeling.

Appendix A

Coding scheme for symbol and concept

Coding source: Transcript of utterances spoken by the stakeholders.

Table A1 LC coding category – coding rules

Category	*Language construct*
Subcategory	CONCEPT
Code	CO
Coding rules	Single coding if noun or set of nouns is identified that: • acts as thedescription of any object or entity from a domain • acts as a complementation of a given CONCEPT
Subcategory	SYMBOL
Code	SY

Coding rules	Single coding if noun or set of nouns is identified that:
	• acts as an unity
	• represents a concept
	• is interchangeable
	and
	• appears for the first time in the situation and
	  • the context contains a definition
	  or
	  • the context contains a demand for a definition
	or
	• replaces a previouslycoded SYMBOL.

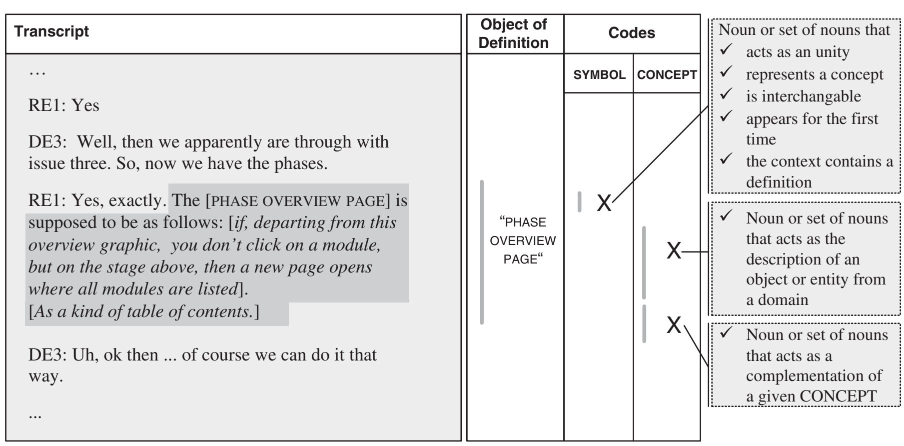


Figure A1 Example for the identification of an OD and the respective LC coding. The OD name is given by the coder and corresponds to one of the observed symbols.

Appendix B

Table B1 Objects of definition and all codes

OD No.	OD English/German	LC Codes and LQ Codes															
1	Administrator	SY	X			X											
		CO	X	X	X	X											
		LQ															
2	Anchor	SY	X														
		CO		X													
		LQ	MLN														
3	Article/Artikel	SY	X			X											
		CO		X	X		X										
		LQ				MLN											
4	Automatic Tagging	SY		X													
		CO	X														
		LQ	INC														
5	Author/Autor	SY	X		X		X	X		X	X						
		CO	X	X	X	X	X	X	X		X						
		LQ	RDC AMB		RDC		RDC AMB	RDC		RDC AMB	RDC						
6	Breadcrumbs	SY		X	X	X	X	X						X			
		CO	X			X			X	X	X	X	X				
		LQ	INC		RDC	RDC		RDC		INC				RDC			
7	Change History/ Änderungshistorie	SY					X										
		CO	X	X	X	X	X	X	X	X							
		LQ	INC														
8	Comment Function/ Kommentarfkt	SY	X														
		CO	X														
		LQ															
9	Contact Management/ Kontaktverw.	SY	X														
		CO	X	X	X												
		LQ															
10	Contact Objects/ Kontaktobjekte	SY				X	X	X	X	X	X	X					
		CO	X	X	X		X	X	X	X	X	X	X				
		LQ	INC					RDC	RDC	RDC	RDC AMB	RDC AMB	AMB				
11	Content Element	SY	X		X					X	X	X					
		CO	X	X	X	X	X	X	X	X	X	X	X				
		LQ			RDC						RDC						
12	Content Element Link/ Content Element Verlinkung	SY			X												
		CO	X	X		X	X	X	X	X	X						
		LQ	INC														
13	Document/ Dokument	SY	X			X											
		CO	X	X	X	X	X	X	X								
		LQ				RDC AMB											
14	Double Log-in	SY	X														
		CO	X	X	X	X											
		LQ															
15	Edit Permission/ Editieren von Inhalten	SY	X		X	X	X				X						
		CO		X	X	X	X	X	X	X		X					
		LQ	MLN		RDC	RDC	RDC				RDC						
16	Edit Protection/ Schreibschutz	SY				X	X	X									
		CO	X	X	X	X	X	X									
		LQ					RDC										
17	Embedded Hyperlink/Eingebetteter Link	SY					X			X							
		CO	X	X	X	X	X	X	X		X						
		LQ	INC														
18	FCK-Editor	SY	X														
		CO	X	X	X	X	X	X									
		LQ															
19	Field Report/ Fallstudien	SY	X	X	X		X		X						X	X	X
		CO	X	X		X	X	X	X	X	X	X	X	X	X	X	X
		LQ		RDC	RDC		RDC		RDC						RDC AMB	RDC	RDC
20	Format Of Input Data/ Eingangsdatenformate	SY	X														
		CO		X													
		LQ	MLN														
21	Full Text Search/ Volltextsuche	SY	X														
		CO		X	X												
		LQ	MLN														

Table B1 Continued

OD No.	OD English/German	LC Codes and LQ Codes															
22	H1 Engine	SY	X														
		CO	X	X													
		LQ															
23	H1 Tag/H1-Tag	SY	X	X			X										
		CO	X		X	X	X										
		LQ	AMB	MLN		INC											
24	Headline Summary	SY	X														
		CO	X	X													
		LQ															
25	Hierarchy/ Hierarchisierung Headline Summary	SY	X			X											
		CO	X	X	X	X	X	X									
		LQ				MLN											
26	History of Requests/ Historie	SY	X														
		CO	X	X	X												
		LQ															
27	Home Button/ Homebutton	SY		X													
		CO	X	X													
		LQ	INC														
28	Homepage/ Startseite	SY		X	X	X											
		CO	X	X	X	X											
		LQ	INC	RDC	RDC	RDC											
29	Integration Of News 1	SY		X													
		CO	X	X													
		LQ	INC														
30	Integration Of News 2	SY		X													
		CO	X	X													
		LQ	INC														
31	Integration Of News 3	SY		X		X											
		CO	X		X												
		LQ	INC			INC RDC											
32	Key Identifier/ Eindeutiger Bezeichner	SY		X													
		CO	X		X	X	X	X	X								
		LQ	INC														
33	Knowledge Platform / Wissensplattform	SY	X						X								
		CO		X	X	X	X	X	X								
		LQ	MLN														
34	Labeling	SY	X														
		CO	X	X													
		LQ															
35	Method.Description/ Methodenbeschreibung	SY	X					X									
		CO		X	X	X	X	X	X								
		LQ	MLN					RDC									
36	MN Relationship/ MN Beziehung	SY	X	X													
		CO			X												
		LQ	MLN														
37	Module Overview Page/ Modulübersichtsseite	SY				X		X		X					X		
		CO	X	X	X	X	X	X	X	X	X	X	X	X	X		
		LQ	INC												AMB		
38	Module/ Modul	SY	X	X		X	X		X		X	X	X				
		CO	X		X	X	X	X	X	X	X						
		LQ		RDC							RDC						
39	Module Link/ Modul-Link	SY	X	X	X												
		CO	X	X	X												
		LQ		RDC	RDC												
40	Morphological Box/ Morphologischer Kasten*	SY		X													
		CO	X														
		LQ	INC														
41	Paper Prototyping	SY	X														
		CO		X	X												
		LQ	MLN														
42	Personalization Concept/ Personalisierungskonzept	SY		X	X												
		CO	X		X	X											
		LQ	INC		RDC												
43	Phase	SY	X														
		CO	X	X	X	X	X										
		LQ															

Table B1 Continued

OD No.	OD English/German	LC Codes and LQ Codes															
44	Phase Overview Page/ Phasenübersichtsseite	SY	X		X	X	X										
		CO	X	X	X	X	X										
		LQ					RDC AMB										
45	Phrase search/ Phrasensuche	SY															
		CO	X	X													
		LQ	INC														
46	Procedure Model/ Vorgehensmodell	SY	X		X	X											
		CO		X		X											
		LQ	MLN		RDC												
47	Questionnaire/ Fragenkatalog	SY	X	X	X												
		CO	X	X	X												
		LQ		RDC	RDC												
48	Red Cloth	SY	X														
		CO	X														
		LQ															
49	Registered User/ Angemeldete Besucher	SY	X	X		X	X	X									
		CO	X	X	X		X	X									
		LQ		RDC AMB		RDC AMB	RDC AMB	RDC AMB									
50	Revision Control/ Versionierung	SY		X													
		CO	X														
		LQ	INC														
51	Role Concept/ Rollenkonzept	SY	X	X													
		CO	X	X													
		LQ															
52	Role/ Rolle	SY	X		X	X	X										
		CO	X	X	X	X	X										
		LQ															
53	Search Function/ Suchfunktion	SY	X		X												
		CO		X	X												
		LQ	MLN														
54	Search Routine	SY	X														
		CO	X	X													
		LQ															
55	Search Suggestions/ Suchvorschläge	SY			X												
		CO	X	X													
		LQ	INC														
56	Short Description/ Kurzbeschreibung	SY	X	X													
		CO	X	X													
		LQ		RDC													
57	Structure P&M/ Struktur Phasen u. Module	SY	X				X	X	X		X	X	X	X	X	X	
		CO	X	X	X	X	X	X	X	X							
		LQ					RDC				RDC	RDC	RDC	RDC	RDC	RDC	
58	Synonym	SY		X													
		CO	X	X													
		LQ	INC														
59	Synonym List/ Synonymliste	SY	X														
		CO		X	X												
		LQ		MLN													
60	Synonym relationship/ Synonymbeziehung	SY	X														
		CO	X	X	X												
		LQ															
61	Toggle Function/ Toggle-Funktion	SY		X			X										
		CO	X	X	X	X											
		LQ	INC				INC RDC										
62	Tracking	SY															
		CO	X	X													
		LQ	INC														
63	Typing/ Typisierung	SY															
		CO	X	X	X												
		LQ	INC		INC												
64	Update Relationship/ Aktualisierungsbeziehungen	SY	X														
		CO		X	X												
		LQ	MLN														

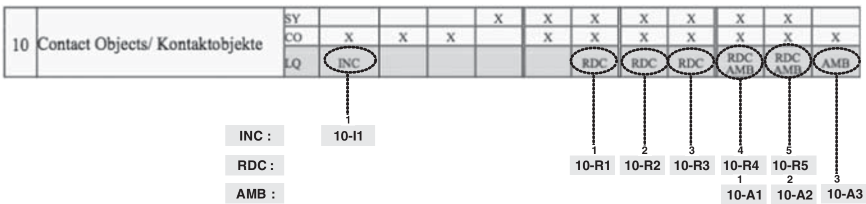
Table B1 Continued

OD No.	OD English/German	LC Codes and LQ Codes																
65	Upload Function	SY	X															
		CO	X															
		LQ																
66	User	SY	X	X		X	X											
		CO	X	X	X	X												
		LQ		RDC		RDC	RDC											
67	User Profile/ Benutzerprofil	SY	X	X		X												
		CO	X	X	X	X												
		LQ		RDC		RDC AMB												
68	Visualization PM/ Visualisierung VM	SY	X	X	X		X			X								
		CO	X		X	X		X	X	X								
		LQ		RDC	RDC	INC				RDC AMB								
69	Volatility Scale /Volatilitätsskala	SY	X		X	X												
		CO	X	X														
		LQ			INC													
70	WYSWYG-Editor	SY	X															
		CO	X															
		LQ																

The double line (ll) stands for the change of situation.
The OD name is given by the coder and corresponds to one of the observed symbols.
Legend: LC codes are in rows 'SY' and 'CO'. LQ codes are in row 'LQ': INC, MLN, RDC, AMB.

Appendix C

Description of the Generation of the Coding ID
Each coding can be identified in Appendix B as follows:



The used coding ID consists of the following parts:

Table C1

OD number (1–70)	Code of the LQ Subcategory (‘I’ = INC, ‘M’ = MLN, ‘R’ = RDC, ‘A’ = AMB)	Coding number for each OD and LQ Subcategory

Example: The ID for the first RDC coding in OD-10 is *10-R1*.

Appendix D

Table D1 Subsets of codings and objects of definition

Deficiencies in Completeness (INC/I)	
All ODs	4, 6, 7, 10, 12, 17, 23, 27, 28, 29, 30, 31, 32, 37, 40, 42, 45, 50, 55, 58, 61, 62, 63, 68, 69.
All codings (all cases)	4-I1, 6-I1, 6-I2, 7-I1, 10-I1, 12-I1, 17-I1, 23-I1, 27-I1, 28-I1, 29-I1, 30-I1, 31-I1, 31-I2, 32-I1, 37-I1, 40-I1, 42-I1, 45-I1, 50-I1, 55-I1, 58-I1, 61-I1, 61-I2, 62-I1, 63-I1, 63-I2, 68-I1, 69-I1.
Subset INC-1	4-I1, 6-I1, 6-I2, 7-I1, 10-I1, 17-I1, 23-I1, 27-I1, 28-I1, 29-I1, 30-I1, 31-I1, 32-I1, 37-I1, 40-I1, 42-I1, 50-I1, 55-I1, 58-I1, 61-I2
Subset INC-1.1	4-I1, 6-I1, 23-I1, 27-I1, 28-I1, 29-I1, 30-I1, 32-I1, 40-I1, 42-I1, 50-I1, 58-I1, 61-I1.
Subset INC-1.2	6-I2, 7-I1, 10-I1, 17-I1, 31-I1, 37-I1, 55-I1.
Subset INC-2	45-I1, 62-I1, 63-I1, 63-I2.
Subset INC-3	12-I1, 31-I2, 61-I2, 68-I1, 69-I1.
Deficiencies in Meaningfulness (MLN/M)	
All ODs	2, 3, 15, 20, 21, 23, 25, 33, 35, 36, 41, 46, 53, 59, 64,
All codings (all cases)	2-M1, 3-M1, 15-M1, 20-M1, 21-M1, 23-M1, 25-M1, 33-M1, 35-M1, 36-M1, 41-M1, 46-M1, 53-M1, 59-M1, 64-M1,
Subset MLN-1	2-M1, 3-M1, 20-M1, 21-M1, 41-M1, 46-M1, 59-M1.
Subset MLN-2	15-M1, 23-M1, 25-M1, 35-M1, 36-M1.
Deficiencies in Non-Redundancy (RDC/R)	
All ODs	5, 6, 10, 11, 13, 15, 16, 19, 28, 31, 35, 38, 39, 42, 44, 46, 47, 49, 56, 57, 61, 66, 67, 68.
All codings (all cases)	5-R1, 5-R2, 5-R3, 5-R4, 5-R5, 5-R6, 6-R1, 6-R2, 6-R3, 6-R4, 10-R1, 10-R2, 10-R3, 10-R4, 10-R5, 11-R1, 11-R2, 13-R1, 15-R1, 15-R2, 15-R3, 15-R4, 16-R1, 19-R1, 19-R2, 19-R3, 19-R4, 19-R5, 19-R6, 19-R7, 28-R1, 28-R2, 28-R3, 31-R1, 35-R1, 38-R1, 38-R2, 39-R1, 39-R2, 42-R1, 44-R1, 46-R1, 47-R1, 47-R2, 49-R1, 49-R2, 49-R3, 49-R4, 56-R1, 57-R1, 57-R2, 57-R3, 57-R4, 57-R5, 57-R6, 57-R7, 61-R1, 66-R1, 66-R2, 66-R3, 67-R1, 67-R2, 68-R1, 68-R2, 68-R3.
Subset RDC-1	6-R2, 6-R3, 10-R1, 10-R3, 10-R5, 15-R1, 15-R2, 15-R3, 15-R4, 16-R1, 19-R1, 19-R2, 19-R4, 19-R5, 19-R6, 19-R7, 28-R1, 38-R1, 47-R1, 47-R2, 57-R2, 57-R3, 57-R4, 57-R5, 57-R6, 57-R7.
Subset RDC-2	5-R1, 5-R2, 5-R3, 5-R4, 5-R5, 5-R6, 10-R4, 13-R1, 19-R3, 35-R1, 38-R2, 39-R2, 42-R1, 46-R1, 49-R1, 49-R2, 49-R3, 49-R4, 66-R1, 66-R2, 66-R3, 67-R1, 67-R2.
Subset RDC-3	6-R1, 6-R4, 10-R2, 11-R1, 11-R2, 28-R2, 28-R3, 31-R1, 39-R1, 44-R1, 56-R1, 57-R1, 61-R1, 68-R1, 68-R2, 68-R3.
Deficiencies in Unambiguity (AMB/A)	
All ODs	5, 10, 13, 19, 23, 37, 44, 49, 67, 68.
All codings (all cases)	5-A1, 5-A2, 5-A3, 10-A1, 10-A2, 10-A3, 13-A1, 19-A1, 23-A1, 37-A1, 44-A1, 49-A1, 49-A2, 49-A3, 49-A4, 67-A1, 68-A1.
Subset AMB-1	10-A2, 10-A3, 19-A1 , 23-A1, 49-A1.
Subset AMB-2	5-A1, 5-A2, 5-A3, 10-A1, 13-A1, 37-A1, 44-A1, 49-A2, 49-A3, 49-A4, 67-A1, 68-A1
The Contingency of Language Quality	
Use of descriptive symbols	See *subset 3* (*Completeness*) and *subset 3* (*Non-Redundancy*).
The Interrelatedness of Requirements	
Cases with codings 'Redundancy and Ambiguity'	(5-R1, 5-A1), (5-R3, 5-A2), (5-R5, 5-A3), (10-R4, 10-A1), (10-R5, 10-A2), (13-R1, 13-A1), (19-R5, 19-A1), (44-R1, 44-A1), (49-R1, 49-A1), (49-R2, 49-A2), (49-R3, 49-A3), (49-R4, 49-A4), (67-R2, 67-A1), (68-R3, 68-A1).
Taking Action for Language Quality	
Explicit actions 'meaningfulness'	ODs: 15, 23, 25, 35, 36.See *subset MLN- 2.*
Explicit actions 'unambiguity'	ODs: 10, 10, 19, 23, 49.See *subset AMB-1* .
Explicit actions 'completeness'	ODs: 40.
Explicit actions 'non-redundancy'	ODs: 10, 11, 19, 28, 38, 49.
No explicit actions 'non-redundancy'	ODs: 5, 6, 13, 15, 16, 31, 35, 39, 42, 44, 46, 47, 56, 57, 61, 66, 67, 68
No explicit actions 'unambiguity'	ODs: 13, 37, 44, 67, 68.