DESIGN, IMPLEMENTATION, AND EVALUATION OF AN ICT-SUPPORTED COLLABORATION METHODOLOGY FOR DISTRIBUTED REQUIREMENTS DETERMINATION

JENS DIBBERN, MICHAEL GEISSER,
TOBIAS HILDENBRAND, ARMIN HEINZL

WORKING PAPER 05/2009
JUNE 2009

Prof. Dr. Jens Dibbern
Institute of Information Systems, University of Bern
Engehaldenstrasse 8, CH-3012 Bern, Switzerland,
jens.dibbern@iwi.unibe.ch

Dr. Michael Geisser
Department of Business Administration and Information Systems,
University of Mannheim, Schloss, D-68131 Mannheim, Germany
geisser@wi.uni-mannheim.de

Dr. Tobias Hildenbrand
Department of Business Administration and Information Systems,
University of Mannheim, Schloss, D-68131 Mannheim, Germany
hildenbrand@wi.uni-mannheim.de

Prof. Dr. Armin Heinzl
Department of Business Administration and Information Systems,
University of Mannheim, Schloss, D-68131 Mannheim, Germany
heinzl@uni-mannheim.de

Working Paper Series in Business Administration and Information Systems

---

University of Mannheim, Department of Business Administration and Information Systems
D-68131 Mannheim/Germany, Phone +49 621 1811691, Fax +49 621 1811692
E-Mail: wifo1@uni-mannheim.de, Internet: http://wifo1.bwl.uni-mannheim.de/

Design, Implementation, and Evaluation of an ICT-Supported Collaboration Methodology for Distributed Requirements Determination

Jens Dibbern
Institute of Information Systems, Department of Information Engineering
University of Bern, Engehaldenstr. 8, Room 204
CH - 3012 Bern
phone: +41 (0)31 631 3889
fax: +41 (0)31 631 4682
email: jens.dibbern@iwi.unibe.ch

Michael Geisser
Dept. of Business Administration and Information Systems
University of Mannheim, L15, 1-6
D-68131 Mannheim, Germany
phone: +49 (0)621/181-1690
fax: +49 (0)621/181-1692
email: geisser@uni-mannheim.de

Tobias Hildenbrand
Dept. of Business Administration and Information Systems
University of Mannheim, L15, 1-6
D-68131 Mannheim, Germany
phone: +49 (0)621/181-1690
fax: +49 (0)621/181-1692
email: hildenbrand@uni-mannheim.de

Armin Heinzl
Chair of Business Administration and Information Systems
University of Mannheim, L15, 1-6
D-68131 Mannheim, Germany
phone: +49 (0)621/181-1690
fax: +49 (0)621/181-1692
email: heinzl@uni-mannheim.de

Design, Implementation, and Evaluation of an ICT-Supported Collaboration Methodology for Distributed Requirements Determination

Acknowledgements

We would like to thank Martin Schader, Christian Becker and Guenther Mueller for instructive comments on earlier versions of this research. We would also like to acknowledge the generous support of the Landesstiftung Baden-Wuerttemberg which had funded the project CollaBaWue out of which this research has been initiated.

About the Authors

Jens Dibbern is a Professor in information systems at the School of Business Adminstration at the University of Bern, Switzerland. He received his Ph.D. in information systems from the University of Bayreuth and habilitated at the University of Mannheim, Germany. His research focuses on various aspects of the division of work in information systems and through information systems, such as outsourcing, offshoring, and design of collaborative systems for supporting distributed work. He has previously published in MIS Quarterly, The Data Base for Advances in Information Systems, Information & Management, WIRTSCHAFTSINFORMATIK, Business & Information Systems Engineering, and others. He is on the editorial boards of MISQ and Data Base.

Michael Geisser is preparing for a spin-off in the field of collaborative event platforms in cooperation with University of Mannheim. He received his Ph.D. in information systems from the University of Mannheim and his research focuses on collaboration in distributed software development – both, in terms of appropriate methods and tool support. In

particular, he published his work on collaborative software development, requirements engineering, and traceability in WIRTSCHAFTSINFORMATIK and pertinent conferences.

Tobias Hildenbrand is a program manager and quality expert at SAP AG as well as associate lecturer at the University of Mannheim. He received his Ph.D. in information systems from the University of Mannheim and was a visiting researcher at UC Irvine. His research focuses on collaboration in distributed software development – both, in terms of appropriate methods and tool support. In particular, he published his work on collaborative software development, requirements engineering, and traceability in WIRTSCHAFTSINFORMATIK and pertinent conferences.

Armin Heinzl is Professor and Chair of Business Administration and Information Systems at the University of Mannheim, Germany. His research and teaching interests are process technologies, information systems outsourcing and collaborative software engineering. He was holding visiting positions at Harvard, Berkeley, and Irvine. His papers have appeared in WIRTSCHAFTSINFORMATIK, Zeitschrift für Betriebswirtschaft (ZfB), Die Betriebswirtschaft (DBW), MIS Quarterly, Information Systems and E-Business Management (ISEB), Journal of Organizational Computing and E-Commerce (JOCEC), and Evolutionary Computation. He is a member of the editorial board of the journal Business & Information Systems Engineering (BISE), WIRTSCHAFTSINFORMATIK and Department Editor of the Zeitschrift für Betriebswirtschaft (ZfB).

Design, Implementation, and Evaluation of a Technology-Supported Collaboration Methodology for Distributed Requirements Determination

Abstract

As information systems development becomes more distributed, information and communication technology (ICT) has become crucial to overcome distance and to enable collaboration between system users and analysts. This study presents the design, implementation, and experimental evaluation of a new technology-supported collaborative methodology for requirements determination. The new ICT-supported methodology enables the elicitation, analysis, specification, and validation of requirements in a distributed environment. Its design follows the theoretical principles of Te'eni's (2001) cognitive-affective model of organizational communication for IT design and combines established methods as well as techniques for requirements identification, formulation, dependency determination, prioritization, and selection in a coherent and innovative way. The resulting prototype is professionally implemented and evaluated in an experiment. The experiment is the first to compare the performance of traditional ways of communication via interviews and document exchange with that of communication via an Internet-based collaboration platform for requirements determination. The results show that, both, the efficiency of the overall requirements determination process as well as the overall quality of the resulting requirements, are higher when using the new collaborative methodology. In terms of quality, the completeness and modifiability of requirements are particularly improved. In terms of efficiency, the user and analyst perspectives need to be distinguished. While the effort for requirements elicitation increases for the analysts, this up-front investment pays off in terms of significantly lower effort for the later specification and validation of requirements. In contrast, the users benefit in particular from lower effort during requirements elicitation and analysis.

Keywords: Requirements engineering, computer-mediated communication, methodology, collaboration tool, experiment, design science, knowledge management system, wiki, partial least squares (PLS)

1 INTRODUCTION

In recent years, information systems development (ISD) has become increasingly challenged by distributed development environments, where both the users (i.e., clients) and the software analysts are locally dispersed (Damian et al. 2003b; Herbsleb et al. 2003). Most prominent for the increasing practice of distributed software development has been the growth in offshoring, where client firms hand over software development work to vendors in low-wage countries (Carmel 1999; Carmel et al. 2005).

The challenge of distributed software development is particularly high in the early stages of the development process, where the initial requirements of a software application need to be determined (Carmel et al. 2005; Herbsleb 2007). Requirements determination includes the elicitation, analysis, specification, and verification of software requirements (Davis 1982; Sommerville 2004). This process can also be described as a process of identifying, extracting, and synthesizing knowledge from various sources (Browne et al. 2001; Byrd et al. 1992; Chen et al. 1991; Vitalari 1985; Walz et al. 1993). On the one hand, there is knowledge about *what* the software should be able to do. This domain knowledge is typically held by the users of the software. On the other hand, there is the knowledge about *how* to design and implement software, which is typically held by the group of software developers. In order to identify, extract, and synthesize the knowledge from these various sources, effective and efficient communication is particularly important (Davidson 2002; Holtzblatt et al. 1995; Keil et al. 1995). Accordingly, overcoming communication barriers has been found to be most critical in requirements determination (Cooper et al. 1979; Curtis et al. 1988; Lyytinen et al. 1987; Macaulay 1996).

This raises the question of how to *effectively* communicate if face-to-face communication, which is a key success factor in requirements definition (Teasley 2002), is no longer feasible on a regular basis (Damian et al. 2008). The answer seems obvious: through information and communication technology (ICT). However, what should such an ICT-tool that supports distributed requirements determination look like? What are the requirements and basic functionalities of such a tool and how should they be implemented?

Previous research has developed a multitude of tools and ICT-supported methodologies for supporting specific activities of requirements engineering. However, the majority of research has concentrated on supporting systems *analysts* in formally specifying functional and non-functional requirements, e.g. through the use of various CASE (Computer Aided Systems Engineering) tools that are augmented with group decision support and electronic meeting functionality (Chen et al. 1991; Dean et al. 1997; Liou et al. 1994; Macaulay et al. 1994). These integrated tools are not specifically designed to support interaction in the preliminary ISD phase of requirements determination, where the focus is on achieving mutual understanding about functional requirements between users and analysts (Leonardi et al. 2008). Moreover, they are not specifically built to support distributed settings, where both users and analysts are locally dispersed. To this end, a number of studies have emerged that focus on enhancing user-analyst collaboration through groupware functionality (Boehm et al. 2001; Lang et al. 2001; Seyff et al. 2005; Sinha et al. 2006).

These computer-mediated collaboration platforms are an important step towards integrating various methods, techniques, and tools for distributed requirements determination, but are deficient in two aspects. First, these integrated methodologies and tools are not theoretically grounded. They are mostly, if at all, based on generic requirements drawn from prior studies or situational, non-generalizable settings and experiences. Second, there is a lack of rigorous evaluation of these new methodologies. Current evaluations are often based on dry runs or case studies that suggest the feasibility of the particular methods and its tool instantiations, but they are deficient in providing evidence for specific improvements in efficiency and effectiveness relative to alternative solutions. Furthermore, these rudimentary evaluations do not explain the resulting and often interacting effects of changes in the methodology, prohibiting a concise feedback into the conceptual or theoretical bases selected.

The purpose of this study is to address these two research gaps; first, to build a novel integrated ICT-supported methodology for distributed requirements determination based on established IT design theory. Such a theory-grounded design would allow to better understand the value of particular design functionalities as well as the contextual assumptions on which the design is based. Since efficient and effective communication plays a central role in requirements determination, we chose Te'eni's (2001) cognitive-affective model of organizational communication for IT design as a theoretical basis for guiding the construction of our novel methodology and its tool instantiation. Second, the newly constructed methodology is tested in an experiment by comparing the performance of traditional ways of communication via interviews and document exchange with that of communication via the newly built Internet-based collaboration methodology. From a methodological point of view, this study draws on the principles of design science (Hevner et al. 2004; Peffers et al. 2008). It addresses the problem of ICT-support for distributed requirements determination, derives the objectives and design requirements from design theory, implements the design and evaluates it through a controlled experiment. The paper is organized accordingly; beginning with a brief review of existing research on ICT supported requirements determination.

2 RELATED WORK

Abstracting from the specific challenge of supporting distributed requirements determination, there is a huge variety of methodologies, techniques, and tools that are directed to support either specific or multiple requirements engineering activities, ranging from requirements elicitation to requirements management (Byrd et al. 1992; Coughlan et al. 2002; Mathiassen et al. 2007; Robinson et al. 2003; van Lamsweerde 2000). In the following, the focus is set on the ICT-supported methodologies (for a literature review and citation analysis, see Appendix A, Table 1).[1]

Early research on computer-mediated requirements engineering has focused on supporting the specification of requirements through CASE tools that help to represent requirements in specific modeling languages (Konsynski et al. 1985; Teichrow et al. 1977). These tools have significantly contributed to improving the mutual understanding among software developers – in particular if extended with support functionality for group interaction (Chen et al. 1991). However, since CASE tools mostly abstract from the natural language of the user domain, they are of limited help for achieving mutual understanding between analysts and users in the initial stage of requirements determination (Guinan et al. 1998; Leonardi et al. 2008). Accordingly, attempts have shifted more towards supporting collaboration and the establishment of mutual understanding between users and analysts. These contributions are often based on established methodologies that aim at supporting closer and more intensive working relationships between users and analysts. For example, Liou and Chen (1994) sought to support the principles of joint application development (JAD) through a group support system (GSS) that helps users and analysts to jointly generate and organize ideas for a new expert system through an electronic brainstorming tool as well as to negotiate and select ideas based on a voting tool. In a like vein, Macaulay et al. (1994) developed a cooperative requirements capturing tool that enables users to discuss, evaluate, change, and agree on certain requirements. Both of these tools, however, are not explicitly designed for distributed requirements determination and rely on ongoing face-to-face meetings as part of their methodology (e.g., enabled through a facilitator). This also holds true for the groupware system Easy WinWin by Boehm et al. (2001). It enables and facilitates heterogeneous

[1] Market-leading commercial RE tools, such as IBM DOORS and RequisitePro as well as Serena RTM and Borland CaliberRM, mostly provide sufficient support for requirements elicitation and analysis, but lack requirements specification and validation capabilities.

stakeholder participation and collaboration during requirements analysis including the partitioning, prioritization, as well as conflict identification and consolidation of requirements. While this tool focuses on the requirements analysis phase, Lang and Duggan (2001) developed a more comprehensive Web-based groupware tool (based on Lotus Notes) allowing to enter requirements in natural language as well as categorizing and prioritizing them. Finally, there are a few solutions that were explicitly designed for distributed requirements determination and management. The first of these tools is ARENA (Anytime, Anyplace, Requirements Negotiation Aids) which aims at extending the EasyWinWin methodology to distributed environments (Gruenbacher et al. 2003; Gruenbacher et al. 2001; Seyff et al. 2005). This tool provides an important basis for distributed requirements analysis; however, it lacks support for requirements elicitation. It also lacks a standard level of usability known from other tools and it does not support synchronous collaboration in the same way as asynchronous work. A relatively comprehensive tool has been developed by an IBM research group (Sinha et al. 2006). Faced with the challenge of supporting the increasingly distributed development work at IBM, they crafted a tool that facilitates the interdependent processes of documenting requirements and holding rich contextualized discussions around requirements. More specifically, their tool provides functionality for synchronous communication among stakeholders, version control, hierarchical composition of requirements, and requirements tracing based on a requirements repository.

In summary, current tools for requirements determination and management provide a considerable variety in functionality, but only few of them have been explicitly designed for *distributed* environments. What is also striking is the limited theoretical grounding of the design functionalities of each tool. The majority of studies jumps right into the description of design objectives, requirements, and functionalities without explicitly referring to some type of design theory (Walls et al. 1992) or theoretical foundations that explain why specific design features are useful for achieving specific design goals. Only few attempts were made to draw on kernel theories or theoretical concepts. As an exception, the design of the Easy WinWin groupware makes reference to the dynamic theory of organizational knowledge creation (Nonaka 1994) arguing that a tool for requirements negotiation needs to support the process of surfacing tacit knowledge from various stakeholders (Gruenbacher et al. 2001). Moreover, Lang and Duggan (2001) refer to human communication theory (Adler et al. 1988) in order to derive their overall design goal. In terms of RE scope, none of the existing approaches is covering and/or integrating all phases equally (cf. Appendix A, Tables 2 and 3).

Second, there is room for improvement regarding the rigorous evaluation of the tools. While some studies make use of practical evaluations in natural settings, e.g. in the form of a beta testing case study (Lang et al. 2001) through feedback analysis from prototype demonstration and trial runs (Seyff et al. 2005) or usability exploration (Seyff et al. 2005), others apply restricted laboratory settings in which usage behavior is observed and documented (Macaulay et al. 1994), or through a student experiment, where the effectiveness of design features is compared (Liou et al. 1994). None of the studies, however, has developed and tested specific hypotheses about expected efficiency and effectiveness improvements through the newly designed methodologies and only one study (Liou et al. 1994) has defined and measured a set of dependent variables for evaluative purposes.

3 DESIGN OF A NOVEL COLLABOARTION METHODOLOGY

The previous section raised the question of how the design of a distributed ICT-supported methodology for requirements determination can be theoretically grounded and eventually improved. To this end, it is beneficial to look beyond the literature on requirements engineering and to recall the main challenge of distributed requirements determination.

3.1 Design Terminology and Methodology

Prior to designing a novel methodology, it is first necessary to clarify the underlying terminology. Over the past years, ISD methodologies have proliferated in great numbers, leading to confusion that partly “stems from the [unclear] notion of ‘method’ or ‘methodology’ […]” (Iivari et al. 2001, p. 186). According to Iivari et al. (2001), “an ISD methodology has been interpreted as an organized collection of concepts, *methods (or techniques)*, beliefs, values, and normative principles supported by material resources (see also Hirschheim et al. 1996; Iivari et al. 1998)). A technique or method “consists of a well-defined sequence of elementary operations” (p.186). Similarly, Avison and Fitzgerald (2006) define an ISD methodology as “a collection of procedures, *techniques*, tools, and documentation aids which will help the systems developers in their efforts to implement a new information system” (p. 24).[2] Hence, the main artifact designed in this paper can be classified as a methodology for requirements determination which includes certain collaborative techniques. Since a methodology is a mental artifact, it is required to instantiate the mental artifact into a technological artifact in the form of an ICT tool which can be applied and evaluated in the field (cf. Hevner et al. 2004)).

3.2 Theory-Based Design Requirements

As noted before, the main challenge of collaboration in this context lies in effective and efficient communication among the stakeholders involved in the elicitation, analysis, specification, and validation of requirements. Accordingly, theoretic models that guide the design of IT for supporting communication may be particularly suited for informing and guiding the design of an ICT-supported methodology for requirements determination. One of the most comprehensive theoretical foundations in this domain is Te’eni’s (2001) *cognitive-affective model* of organizational communication for IT design. Te’eni’s model prescribes

---

[2] Within the closely related field of Software Engineering (SE), Sommerville defines a SE method as “a structured approach to software development whose aim is to facilitate the production of high-quality software in a cost-effective way” including “a number of different components [such as] system model descriptions, rules, recommendations, and process guidance” (p. 12) where a software process “is a set of activities that leads to the production of a software product” (p. 64). These activities correspond to the “elementary operations” mentioned by Iivari et al. (2001, p.186) or the “subphases” (Avison and Fitzgerald, 2006, p. 24).

what types of communication processes (broken down into communication goals, communication strategies, communication medium, and communication form) are most suitable given certain types and instances of communication inputs (such as task characteristics and sender-receiver attributes) for achieving a high communication impact (i.e. a mutual understanding and good relationships among the communicating actors).

Since achieving mutual understanding between users and analysts is one of the key outcomes of the requirements determination process, Te'eni´s model represents a preferable and comprehensive framework for guiding the design process. Based on this framework, the collaboration methodology's underlying kernel theory is outlined in the following section. Thereafter, the concrete instantiation of our collaboration methodology is represented in the form of procedural guidelines and ICT support.

Before applying Te'eni's (2001) model, some preliminary thoughts about the role of communication in requirements determination are necessary. The communication process of requirements determination takes on a dual role. On the one hand, communication is required among and between users and analysts for achieving mutual understanding about a set of requirements that are usually documented in written form (Valusek et al. 1987). For this purpose, different types of communication channels may be used, such as telephone, e-mail, or chat. On the other hand, the outcome of the communication process (i.e. the explicated and validated software requirements specification) may become a communication medium itself. The requirements document represents a preliminary design. In analogy to the function of a prototype (Lichter et al. 1994), it serves as a communication basis capturing the knowledge of all stakeholders that contribute to its design (Grunwald et al. 2007). As far as the contribution of each particular stakeholder to the requirements document is made visible to the other parties, the requirements document serves as a communication medium that facilitates the exchange of knowledge and the transparency of the determination process.

Taking this duality of communication in requirements determination into account, the IT design model by Te'eni (2001) is applied subsequently. It begins with looking at the communication inputs for requirements determination.

3.2.1. Communication Inputs

Communication inputs can be characterized by three elements: the characteristics of the task situation, the (communication) sender/receiver distance and the surrounding (cultural) values and norms (Te'eni (2001)). Our emphasis in this context is on the characteristics of the requirements determination tasks. The other two input elements, cognitive and affective distance as well as values and norms depend largely on the situation in which the technology will be used and are, thus, difficult to assess in advance. In contrast, the characteristics of the tasks are assessable in advance and may be considered as a crucial input factor for the communication process. They can be described along three distinct characteristics: task analyzability, task variety, and task temporality.

Task analyzability. The requirements determination process is often characterized by a relatively low task analyzability, since it is difficult for a particular user or analyst to determine how exactly the task should be accomplished. For example, it is impossible to determine *a priori* who should speak with whom and when as the contributions of each stakeholder can hardly be anticipated. There are various sources of requirements process uncertainty (Davis 1982) so that even analyzing how a particular requirement was generated (source traceability) ex post can be quite challenging (Sommerville 2004).

Task variety. In contrast, task variety seems to be rather low at a first glance. On an abstract level, it is well known what kind of tasks need to be accomplished, such as the requirements elicitation, analysis (including negotiations and selection), specification, and validation. But on the distinct operational level of action within these subtasks, task variety increases substantially. For instance, there are many ways and methods, an analyst can elicit requirements. For instance, s/he may apply individual interviews, focus group interviews, survey techniques, brainstorming or protocol analysis, just to name a few, which indicate a high task variety on the subtask level. Other examples to support this notion are requirements specification and validation. Requirements may be specified in plain text, structured documents or with the help of conceptual modeling language. The number of conceptual modeling languages which incorporate software requirements has significantly increased during the past decades. Even the de facto modeling standard UML 2.0 offers different techniques for specifying the requirements of IS. Use cases, state charts, activity diagrams, and object models are just a few examples which demonstrate the task variety that analysts and users have to cope with. For requirements validation, different approaches like inspections, reviews, generation of test cases, or visual validation can be named. For this reason, the task variety in the requirements determination process can be seen relatively high.

Task temporality. Tight project schedules combined with a high amount of interdependent tasks often require the provision of quick feedback among stakeholders, leading to time related demands to complete the task and to move to the next task (Herbsleb et al. 2003). Quick response is proposed to be particularly important in distributed environments (Cramton 2001), thus, *temporality* is rated quite high.

The low task analyzability, its high variety, and its high temporality lead to a *high communication complexity*. Requirements determination is characterized by high reciprocal interdependence of communication and action since there is interdependence between requirements of various stakeholders (i.e., requirements may overlap, complement, or contradict each other). This requires constant feedback mechanisms among users as well as between users and analysts for ensuring that requirements are correctly understood and comprehensible (Valusek et al. 1987). Moreover, due to the multiplicity of views and preferences held by the communicators, some sort of mechanism is required to align the different goals and perspectives to achieve a common outcome. This need for influencing and managing interdependencies, amount to a relatively *high cognitive complexity*. Furthermore, requirements determination often demands users and analysts to work in parallel and under pressure, i.e. different users may explicate their requirements at the same time, which leads to *high dynamic complexity*. Finally, requirements determination is also often characterized by *affective complexity*, since users and analysts often have different mental models and dispositions as well as personal interests which most likely lead to misunderstandings and conflicts. Indeed, avoiding and managing conflicts is one of the main challenges during requirements analysis (van Lamsweerde et al. 1998).

Taken together, a distributed requirements methodology needs to be able to cope with high cognitive, high dynamic, and varying situational affective complexities which are induced by the low analyzability, high variety, and high temporality of the corresponding tasks (see Appendix B, Table 1).[3] Whilst considering these complexities, a careful analysis of the communication process is required and will be taken up next.

3.2.2. Communication Process

According to Te'eni, the communication process is triggered by the communication goals of the participating actors, who in turn, chose different communication strategies in order to cope with the communication complexity. The strategy itself determines the communication medium and the message form which may have reciprocal effects on each other as well as repercussions on the communication strategy.

The major communication goals are to mutually "instruct" software analysts and users about the desired and useful functionality of a software product, to manage the interdependent actions that occur during the elicitation, analysis, specification, and validation of software

---

[3] Affective complexity will not less emphasized at this point, since it can substantially vary between software projects due to shared experiences of the involved parties Guinan, P.J., Cooprider, J.G., and Faraj, S. "Enabling Software Development Team Performance During Requirements Definition: A Behavioral Versus Technical Approach," *Information Systems Research* (9:2) 1998, pp 101-125..

requirements, as well as managing and fostering the relationship of the participating actors in such a highly complex setting. During the process of instruction, managing interdependent action and managing relationships, the ability to influence the receiver according to sender's (communication) intentions becomes crucial. Only if sender and receivers, whose roles are steadily changing, are able to positively influence each other, a mutual understanding can be achieved. Without mutual understanding between actors, the determined requirements are not likely to deliver the necessary condition for the future success of the system.

3.2.2.1. Communication Strategy

The communication strategy draws on the communication goals and courses of action for achieving these goals. According to Te'eni, it consists of one or more of the following elements: the level of required contextualization, affectivity, control through testing and adjusting, control through planning, perspective taking, and attention focusing. As indicated, a distributed requirements methodology needs to be able to cope with high cognitive and dynamic complexity.[4] This requires the choice of an appropriate communication strategy which is taken up next.

Contextualization describes the provision of explicit context in the communication messages. Contextualization should be fostered, since the possibility of misunderstandings is conceivably high due to the high cognitive complexity and low analyzability.

> *"The strategy of building context into the message decreases the probability of misunderstanding and, thereby, increases the probability of accomplishing the goal of thinking collectively." (Te'eni 2001, p. 269)*

Moreover, the presence of multiple perspectives, as it is the case for users and analysts, calls for high contextualization (Katz et al. 2007). Contextualization helps users and analysts

[4] Affective complexity will not be further considered at this point, since it can substantially vary between software projects due to shared experiences of the involved parties Ibid.). which makes it difficult to assess, too.

to make sense of requirements by adding and linking related information, such as indicating relationships to other requirements (requirements traceability, Sommerville 2004), showing the original problems that caused the requirements, providing details about requirements, allowing to trace changes of requirements over time, knowing their originators, and thus increasing awareness (based on Majchrzak et al. 2005b, p. 111).

Affectivity expresses the provision of affective elements in messages. Those elements can be (negative) emotions and moods which can be used to draw attention as well as to inform, influence or motivate other actors.

Control by testing and adjusting should be enabled since it is a viable strategy for coping with high dynamic complexity and high task temporality. This can be achieved through constant feedback loops between users and analysts, in which changes in the requirements documents are examined (i.e. tested) by each other.

Control by planning describes whether patterns of communication and contingencies are formalizable ahead of the communication process. It is considered to be rather low if dynamic complexity and temporality are high, since it is difficult to foresee the content of particular ideas (i.e. requirements) and to anticipate who is communicating what and when. Thus, any too narrowly focused procedure or content specific template would be rather counterproductive (Flynn et al. 1994).

Perspective taking should also be supported in order to cope with the high cognitive complexity of the communication process within requirements determination. It is important that the system analysts are able to "step into the shoes" of the users, which requires them to publicly observe the actions of the users, e.g. by observing what changes users make to the requirements documents and how they evaluate particular requirements. In this vein, Boland et al. (1994, p. 467) argue that information systems for distributed cognition should support *multiplicity*, i.e. the ability for "actors to compare and contrast interpretations" that can be made by different users and analysts in parallel.

Finally, *attention focusing*, which refers to directing or manipulating the receiver's processing of a message, should be supported since it is necessary in cases of high communication complexity. Moreover, the high risk of information overload due to the contributions of various stakeholders (Te'eni 2001, p. 268) often requires some type of highlighting, summarizing, sense making, visualization, and information aggregation by the analysts in order to systemically analyze and negotiate requirements with the users (Spence 2007).

Tables 2 and 3, appendix B, summarize the attribute values of the communication strategy elements when considering communication complexities and communication inputs. It is clearly visible that contextualization and control by testing and adjusting are the most important communication strategies. Furthermore, perspective taking and affectivity play also a vital role as communication strategy parameters. They determine the communication medium and form. Both will be taken up next.

3.2.2.2. Communication Medium and Message Form

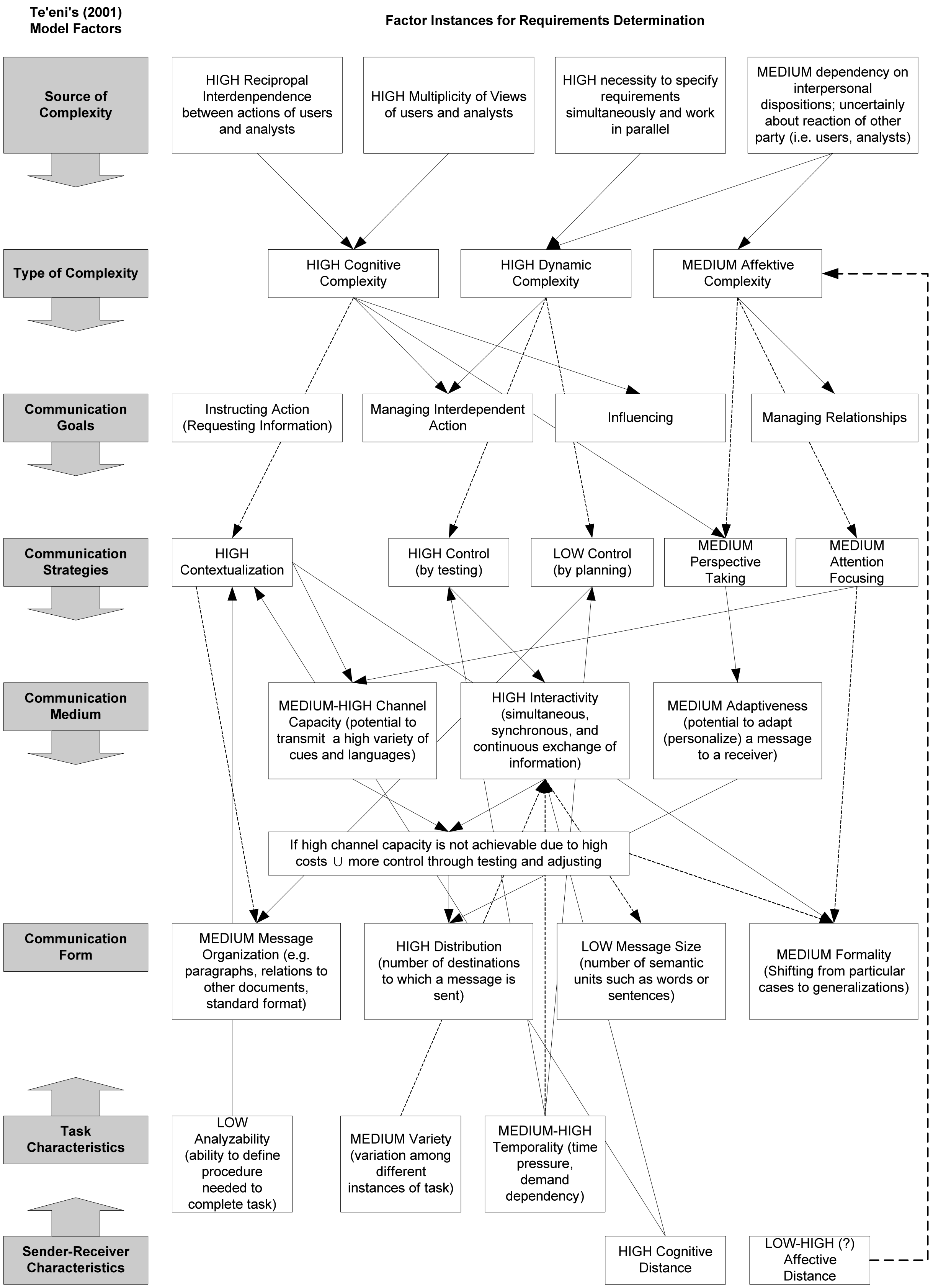
Having identified relevant communication strategy elements for requirements determination such as high contextualization, high control through testing, low control through planning, as well as relatively high perspective taking and attention focusing, the major elements of the communication medium and message form that should be supported by an ICT-enabled methodology can be derived accordingly (Te'eni 2001).

Communication Medium. For differentiating forms of communication, Te'eni refers to media richness theory (Daft et al. 1986) distinguishing between channel capacity (potential to transmit a high variety of cues and languages), degree of interactivity (simultaneous, synchronous, and continuous exchange of information), and level of adaptiveness (potential to adapt, i.e. personalize, a message to a receiver). Given the properties of the communication strategy identified, a medium with high channel capacity, high interactivity, and high adaptiveness may be particularly suited. However, in situations where high channel capacity, such as face-to-face communication, is infeasible due to cost or technology constraints, a compensation strategy is required, such as more control through testing and adjusting (Te'eni 2001, p. 275). Since frequent face-to-face communication is infeasible in distributed requirements determination, testing and adjusting mechanisms enabled through ICT are particularly important. At the same time, interactivity and adaptiveness should also be enabled for enhancing control through testing and adjusting (high) and perspective taking (high) respectively. The communication form attributes are summarized in Tables 4 and 5, appendix B.

Message Form. In terms of message form, Te'eni distinguishes between message organization, distribution, size, and formality. Due to the required high level of contextualization along with the low level of control through planning, a medium level of *message organization* may be most suited. Thus, the requirements document may need to be structured in a certain way and, therefore, it may be useful to represent the hierarchical order of requirements and their dependencies, e.g. in the form of semi-structured documents. In other words, organization is necessary for coping with high cognitive complexity, however, not at the cost of restricting spontaneous input and creativity of users as well as analysts (Herbsleb et al. 1999). Moreover, *message distribution* should be high, so that all stakeholders are informed about the current state of the requirements documents. While it is necessary to ensure transparency about changes of requirements, at the same time information overload should be avoided (Kotonya et al. 2004). More distribution calls for more organization. Users and analysts need to bring order into diverse pieces of information. Moreover, due to the required high level of interactivity, *shorter rather than longer messages* are preferable. In particular with computer-mediated communication, long messages may be counterproductive (Trevino et al. 1987). Finally, *message formality* is more difficult to determine. It refers to the extent to which it is desirable to advance from more specific cases to more general abstractions. In requirements determination, there certainly is a necessity to synthesize and organize diverse requirements into some higher level requirements formats at an aggregate, generalized level. However, under specific circumstances, it is often necessary to model the specific needs of particular users on a disaggregated, specialized level. Thus, the high need for contextualization calls for less formality, while control (through testing and adjusting) and message organization call for more formality. This apparent trade-off is alleviated by taking a dynamic perspective. It may be argued that during requirements elicitation and analysis, high contextualization and less formality are necessary, while later on, when it comes to requirements specification and validation, more formality is required (Westrup 1999). Tables 6, 7, and 8, appendix B, summarize the attribute values discussed.

The entire reasoning according to Te'eni's cognitive-affective model is visualized in figure 1.

Figure 1: Design elements and attribute values according Te'eni's theoretic model



3.3 Design and Implementation

Based on the preceding analysis, it can be concluded that a new ICT-supported methodology for distributed requirements determination should realize the conceptual aims of high contextualization, control through testing, perspective taking, and attention focusing. A suitable communication form should entail a medium level of message organization, high message distribution among stakeholders, relatively low message size, and a medium level of message formality. It should be taken into account, however, that requirements determination is a dynamic process consisting of iterative stages, such as elicitation, analysis, as well as specification and validation. During this process, the requirements for ICT-support are slightly shifting. While in the early stages it is necessary to enable spontaneous communication and rich information gathering associated with high contextualization, later stages require more information aggregation and organization as well as more formality. By linking the outcome of the later stages to the outcome of earlier stages, a comprehensive picture should be provided and maintained enabling systems analysts to move from high abstraction in requirements specification documents to more detailed descriptions. It is therefore necessary to develop an integrative tool that combines different functionalities, such as assisting in the generation of requirements documents, organizing requirements, exploring of alternatives, extracting relevant portions of it, maintaining both source traceability links to subsidiary elicitation material and forward referencing capabilities as well as integrating a variety of interactive presentation media - e.g., graphics, originals of documentation (van Lamsweerde 2000, p. 11).

Technology Platform. Combining different tools and methods into a coherent ICT-supported methodology requires a common collaboration platform. Since the output of the requirements determination process is dynamically linked to subsequent phases in the software development life cycle, such as design, programming, testing, and continuous requirements management (Egyed et al. 2004), we chose *CodeBeame*r by Intland Software as a professional Java-based collaborative software development platform (CSDP). Compared to other CSDP, it provides a comprehensive base functionality (e.g. regarding documentation, change, and workflow management, as well as issue tracking and information interlinking; Robbins 2005), a Web service-based application programming interface (API) as well as plug-ins for the established software development platforms Eclipse and NetBeans (Rodriguez et al. 2007; Sinha et al. 2006). In the following, the key add-on support functionalities that were specifically designed for supporting the main stages of requirements determination are introduced.

3.3.1. Support for Requirements Elicitation

Initial Meeting. Empirical studies have shown that even if the majority of requirements is determined in a distributed environment, it is important that users and analysts know each other personally in the first place (Edwards et al. 2005). For this purpose, an initial personal meeting, facilitated by a moderator, was found to be particularly effective, since it enables the generation of interpersonal trust (Damian et al. 2003a; Damian et al. 2003b). Accordingly, it was decided to include such an up-front meeting as the first stage in our new methodology. With this personal meeting, the *affective distance* among users and analyst is reduced (Orlikowski 2002) which in turn mitigates the risk of low communication frequency in distributed requirements determination (Herbsleb et al. 2003; Te'eni 2001).

Wiki-Enabled Requirements Identification and Formulation. In the course of identifying and formulating requirements, it is particularly important to enable informal and spontaneous communication that is rich in contextualization and perspective taking (Damian et al. 2003b). For this purpose, a wiki was integrated into the collaboration platform.[5] A wiki is an Internet-based asynchronous collaborative hypertext authoring system that enables

---

[5] The wiki engine of the open source project JSPwiki was used (http://jspwiki.org/ (July 7, 2008) since it is also based on Java technology (cp. codeBeamer platform).

everyone with access to incrementally and rapidly add, change, or delete text contained in interlinked web pages, so-called “articles”. An essential characteristic of a wiki is the possibility to collaboratively edit and transparently refine the content created by others (Ebersbach et al. 2005; Leuf et al. 2001). Thus, the utilization of a wiki nurtures many-to-many communication and co-authoring of requirements documents (i.e., in the form a wiki article). Moreover, there is the possibility to write comments pertaining to particular requirements and to discuss certain requirements in an associated discussion thread or forum. These comments and discussions are pervasively linked to particular (requirements) documents, and hence continuously provide context information. Using a wiki as knowledge and collaboration platform is especially valuable in situations where it is important to quickly aggregate input from different users at different times (high dynamic complexity, Decker et al. 2007). While unlike e-mailing, no specific link exists between sender and receiver of externalized knowledge, thus reducing interaction intensity, at the same time a wiki allows for a higher level of *perspective taking* by enabling users and analysts to publicly see new edits and changes of each other. *Awareness* is further enhanced through automatic e-mail notification when new requirements are added or existing ones are changed (Herbsleb et al. 2003), while the actual requirements documents are centrally captured on the wiki article repository server. Moreover, the interaction processes are captured on the history page and/or discussion page associated with each requirements document. Additional knowledge integration functionalities that further increase *contextualization* are version control, links between wiki pages and to other project resources, categorization, a search function, and a glossary to enable a common understanding of specific terminology of users and analysts. The wiki utilized here is also extended for embedding scalable graphics, incurring content from Word documents, and locking certain areas of the requirements document so that different stakeholders can work on different parts of the same document concurrently (see Figure 2 for an example).

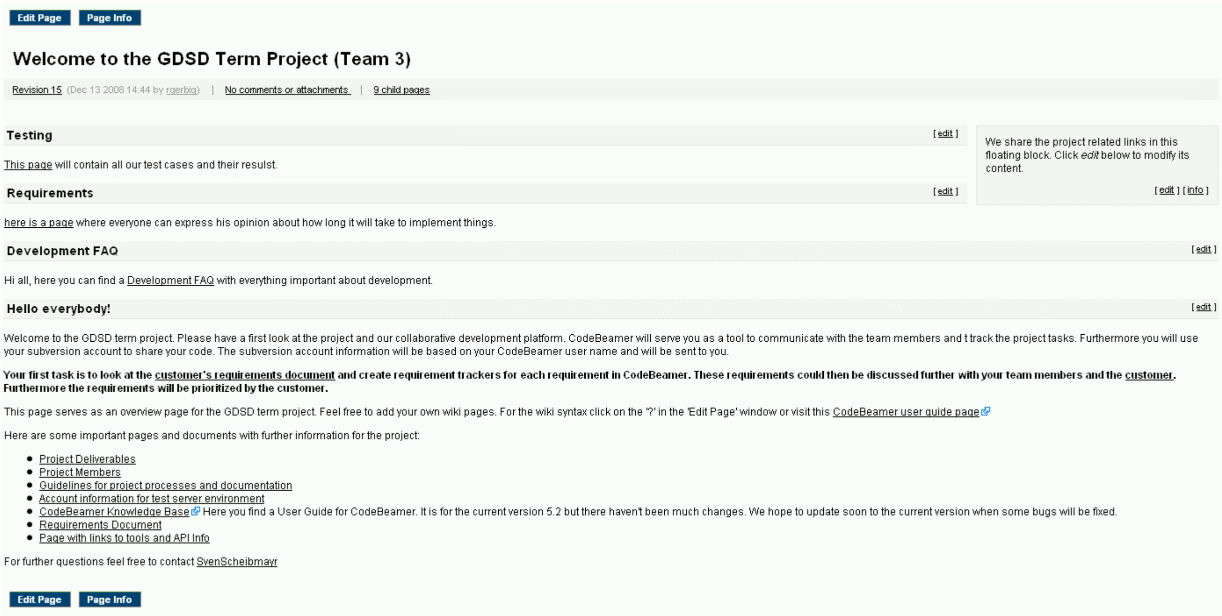


Figure 2: codeBeamer's Wiki-Based Requirements Elicitation

3.3.2. Support for Requirements Analysis

The result of the Wiki-based requirements elicitation is a set of requirements documents that are categorized, associated with diverse comments and discussions, and partially interlinked. The question is raised, however, whether all of these requirements are indeed equally useful and what kind of dependencies exist between requirements. It has been witnessed in different wiki projects that it is very hard to maintain a good structure in a traditional wiki system when the content grows (Decker et al. 2005; Klein et al. 2005). Accordingly, traditional wikis, which are intended to edit and display plain text, exhibit the drawback that they do not support the development and maintenance of a structure within the wiki system and its articles. Hence, there is a need for structure, e.g. for specifying requirements dependencies as well as for distinguishing and selecting those requirements that are most useful.[6]

---

[6] In addition to semi-structured wiki templates, codeBeamer's wiki engine has been extended by means of semantically enriched inter-wiki links, i.e., a short notation for referencing other project resources.

When it comes to determining interdependencies between documents, which is the responsibility of the analysts, a centralized solution is opted for (Damian et al. 2008). Determining dependencies between requirements is an important preliminary step prior to requirements selection, since it is often infeasible to select particular requirements in isolation if they depend or build upon the implementation of other requirements. For this purpose, a method for determining so-called *requirements sets* was developed based on vectored graph representations. The requirements sets are defined by the analysts resulting in a matrix that shows the interdependencies between requirements and can be visualized for analysis purpose.

For requirements selection, a combination of a centralized and decentralized solution based on the *cost-value*[7] *approach* was chosen, since it is established for providing quantifiable decision support for requirements analysis (Karlsson et al. 1997). The business or customer *value* of requirements is determined *decentrally* through pairwise comparisons of requirements by the users implemented by a PHP-based questionnaire tool. The voting procedure and consolidation algorithm is based on the multi criteria decision method "Analytic Hierarchy Process" (AHP, Saaty 1980), which is also incorporated in the Easy WinWin approach (Ruhe et al. 2002; Ruhe et al. 2003). The *costs* in terms of person-days and months are centrally estimated by the analysts; however, this estimation procedure can also be conducted among distributed analysts and developers.

Finally, the cost-value ratio for each requirement is calculated and graphically depicted (as can be seen in Figure 3). Whenever a requirement is dependent on other requirements, the whole requirements set is aggregated and graphically depicted with its overall cost-value ratio

---

[7] Within the context of requirements determination, the terms "value" and "utility" can be applied interchangeably, since both describe the "importance or usefulness" (cf. Longman Dictionary of Contemporary English) of particular requirements Nunamaker, J., Briggs, R.O., De Vreede, G.-J., and Sprague, R.H.J. "Enhancing Organizations Intellectual Bandwidth: The Quest for Fast and Effective Value Creation," *Journal of Management Information Systems* (17:3) 2000, pp 3-8.. Hence, value and utility pertain to the quantitative prioritization as part of requirements analysis activities (see also Boehm 2003).

shown as well. The representation of dependencies together with the representation of the cost-value graphs can be regarded as an *influencing* mechanism that supports *attention focusing* by visually highlighting the relative usefulness of certain requirements. The graphical representation also helps to achieve a higher level of *distribution* since the hierarchical composition and relative advantage of requirements become visible to all stakeholders. Moreover, *formality* is supported, since the selection procedure is made transparent to all stakeholders, helping to achieve an "organizationally accepted representation of action" (Te'eni 2001, p. 277) and maintaining design rationale even in very early project stages (Dutoit et al. 2006). Finally, the use of *mixed (message) form* (such as text and graphics) also directly contributes to better mutual understanding and awareness since it acknowledges that actors differ in their preferred mode of expressions (Boland et al. 1994).

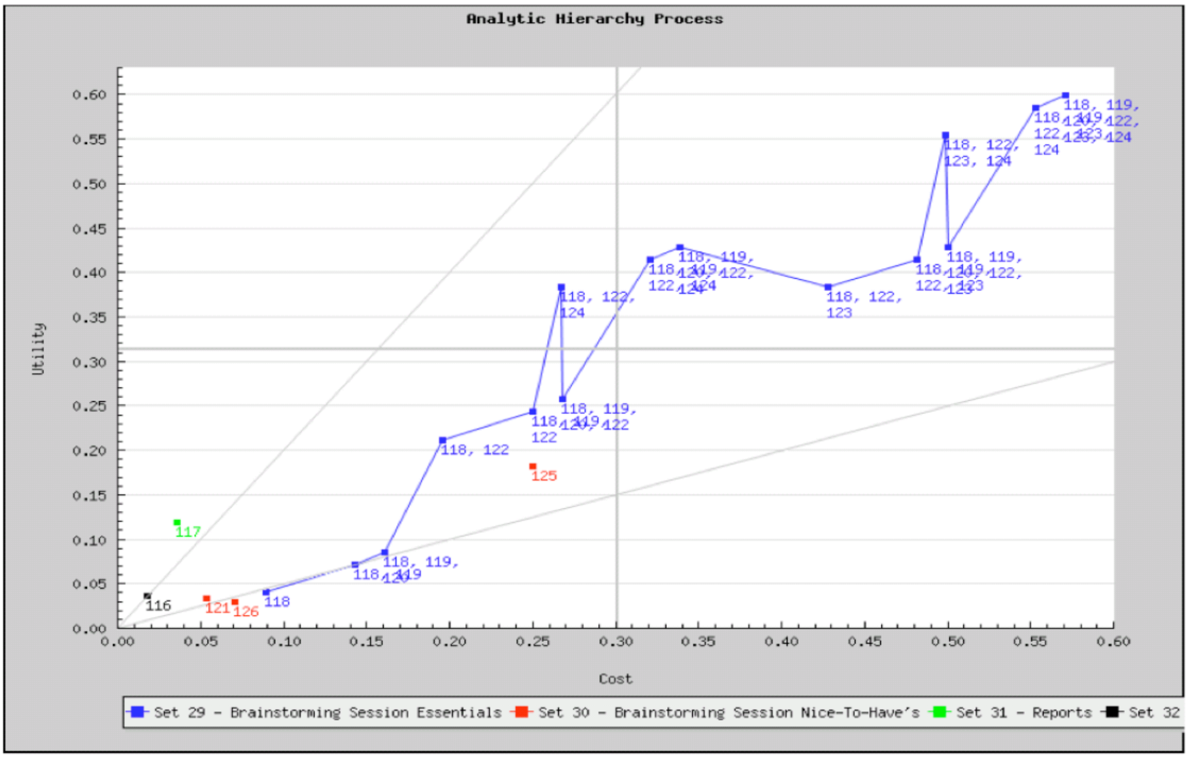


Figure 3: Cost/Value Analysis including Requirements Sets

3.3.3. Support for Requirements Specification and Validation

What follows is the more detailed and formal specification of the selected requirements as well as their validation. For the purpose of specification, the VOLERE template, i.e. so-called *snowcards* were chosen (Robertson et al. 2005; Robertson et al. 2006; Robertson 2007) and integrated into both, the wiki through its template function and the codeBeamer's issue tracker. Thus, requirements selected from the wiki specification are automatically transformed into the new snowcard template and exported to the tracker system while adopting already existing context information, such as name, description, dependencies, value, and cost (cf. Figure 4). These templates help the analysts to organize and extend requirements. For example, there is a requirement reasoning field which forces analysts to step into the shoes of the users and summarize the main *rationale* for having a particular requirement implemented (Dutoit et al. 2006). Since the newly specified requirements are available in the wiki, the users can immediately validate them through textual annotations, while the respective tracker template allows for annotations and continuous user feedback throughout the process. This rather informal specification and validation procedure is in line with the previous theoretical analysis which prescribed a medium level of organization and formality (see also Damian et al. 2003b; Flynn et al. 1994; Nosek et al. 1988).

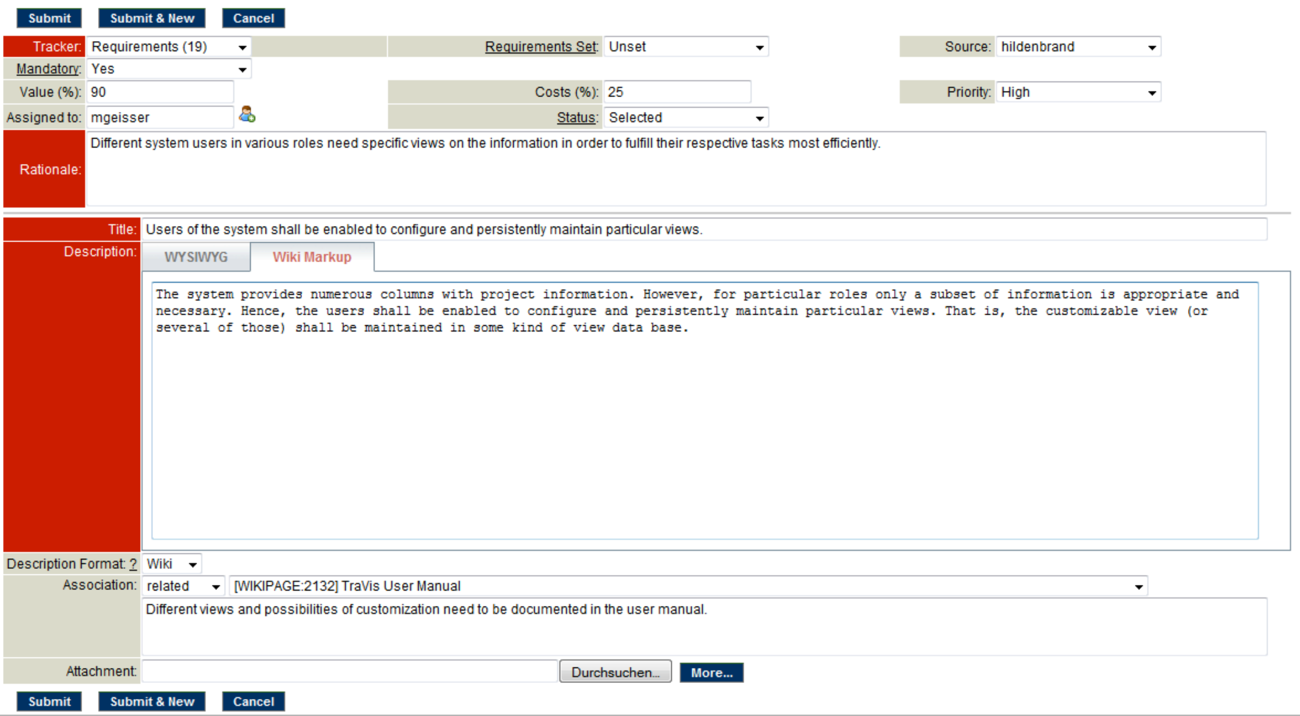


Figure 4: codeBeamer's Tracker for Requirements Specification

3.4 Summary and Comparison with Traditional Approach

In order to understand the new ICT-supported methodology in its entirety, it is useful to compare it with established alterative procedures. A number of studies have shown that the interview technique is one of the most widely accepted techniques for requirements determination (Holtzblatt et al. 1995; Keil et al. 1995; Neill et al. 2003; Vitalari 1985). That is, the analysts seek to extract the necessary information from the users by asking more or less structured questions on desired functionality (Marakas et al. 1998), documenting the requirements, and sending the specifications back to the users for validation. If face-to-face meetings are infeasible on a regular basis – as it is the case for distributed environments – the most straight forward approach is to substitute it with the second richest communication channel which is, according to media richness theory, audio and video-conferencing. While distributed requirements determination aligns such verbal synchronous exchanges with other asynchronous channels, such as rich text based media (Damian et al. 2008), for the purpose of keeping the methodologies distinctive and comparable, no mixture of verbal and non-verbal communication for a particular methodology is preferred in this study (except for the exchange of documents via e-mail). Thus, the traditional approach considered here is mostly characterized by verbal synchronous communication along with document-exchange via e-mail. Table 1 summarizes both approaches and compares the traditional and the novel approach for each of the main requirements determination stages. Figure 5 illustrates the novel process steps of the novel methodology (cp. also Figure 1, Appendix C).

Table 1. Comparison of Methodologies for Distributed Requirements Determination

Phase	***Traditional Methodology*: Communication via inter-views and document exchange**	***Novel Methodology*: Communication via internet-based collaboration platform**
Requirements elicitation		
Initial personal meeting and determination of moderator	YES	YES
Requirements identification and formulation	1. Analysts conduct interviews with users 2. Analysts explicate requirements in document (e.g. MS Word)	Analysts and users collaboratively create requirements documents in form of wiki pages using collaboration platform

Requirements analysis		→ wiki pages are imported into evaluation tool
Requirements dependency determination	Analysts write dependencies into Word document	Analysts create matrix of dependencies with evaluation tool
Requirements prioritization		
- Cost estimation	Analysts write costs into Word document	Analysts use tool for cost assignment to requirements
- Value estimation	1. Analysts conduct interviews with users 2. Analysts write value into Word document	Users use tool for pairwise value comparison based on AHP functionality
Requirements selection based on cost-value comparison	Discussion between analysts and users in personal meeting (e.g. through video conferencing)	Graphical representation of cost-value ratio (including consideration of dependencies) allows quasi-automatic selection
Requirements specification and validation		→ Automatic import into requirements tracker (from evaluation tool to platform)
Requirements specification including organization and extension (e.g. rationale for each requirement)	Analysts structure requirements document using Word template (based on VOLERE)	Analysts structure each requirement using requirements tracker (based on VOLERE)
Requirements validation	Users receive structured Word document via e-mail and provide comments	Users access collaboration platform in parallel to specification and provide wiki-based comments

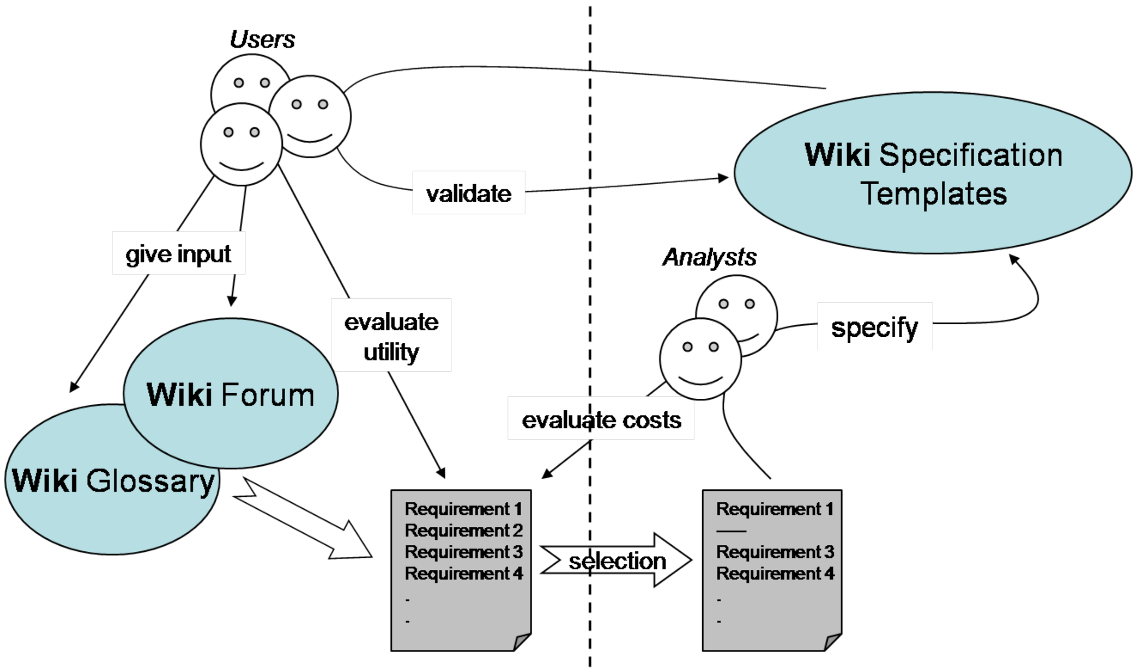


Figure 5: Process Overview of Novel Collaboration Methodology

4 EXPERIMENTAL DESIGN

In order to evaluate the new technology-supported collaboration methodology, an experiment in the form of replicated software projects was conducted with one factor and two treatments (Wohlin et al. 2000). The treatments correspond to an experimental group of development teams which apply the novel methodology and control group teams which use the traditional methodology (Kitchenham et al. 2002). Notably, a pilot study with students from Puerto Rico and customers from Sweden and Japan was conducted prior to the experiment in order to evaluate the methodology and respective tool support in terms of applicability.

4.1 Goals and Hypotheses

The objective of the experiment is to test the efficiency and effectiveness of the new methodology against the traditional one. That is, the two alternatives represent the instances of the independent variable "type of methodology" which is proposed to impact two dependent variables, efficiency and effectiveness. Efficiency refers to the personnel costs incurred in determining requirements broken down into a user and an analyst component. Effectiveness refers to the quality of the outcome of the requirements determination process, i.e. the specified requirements.

In terms of *effectiveness*, a number of empirical studies on distributed software development have shown that the quality of software requirements is particularly important for overall project success (Heeks et al. 2001; Krishna et al. 2004; Nicholson et al. 2001). According to the ”Practices for Software Requirements Specifications“ recommended by the IEEE Computer Society (IEEE 1998), high quality requirement specifications have to be correct, unambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable, and traceable (for a definition and operationalization of each metric, see Section 4.5).

Based on the preceding discussion, it is proposed that these objectives are better achieved with the novel ICT-supported collaboration methodology. Since it is specifically designed for supporting *distributed* requirements determination, there should be a better fit between the means of communication and the requirements determination task leading to better team performance (Zigurs et al. 1998).

The enhanced support capability for contextualization should lead to more *complete* requirements, e.g. due to the possibility to establish links to more detailed descriptions. The possibility to collaboratively work on requirements increases mutual awareness and perspective taking which leads to higher control capacity (i.e. the peer review principle: four eyes see more than two, Damian et al. 2003b) and hence improves *correctness*. Moreover, *unambiguity* (i.e., that each requirement ideally allows only for one interpretation) should be enhanced in order to support high contextualization and perspective taking during the process of requirements determination. This allows for the consideration of multiple interpretations, which is a precondition for reaching consensus about one single interpretation. In addition, tool support for the explicit highlighting of dependencies and the rationalization of the selection procedure should lead to less ambiguity. Enhanced contextualization, e.g. by means of using a wiki-based glossary, is also expected to lead to both, less ambiguity and higher *consistency*, i.e. that terms are used and understood equally among stakeholders. *Modifiability* should be improved, since adapting requirements requires the analysts or users to be aware of existing requirements and to be able to trace them. Furthermore, the automated export of documents from the wiki into the requirements tracker facilitates the atomic representation of the requirements content and, thus, also increases modifiability. Finally, *verifiability* may be improved in that the multiplicity of views puts higher demands on users and analysts to exactly specify what they mean, while providing extra decision support. This subsequently helps to better control how well particular requirements are covered by the software application, e.g. for validation or status reporting purposes. Taken together, this leads to the following hypothesis (see H1 in Figure 6):

Hypothesis 1: *Compared to the traditional interview- and document-based methodology, the novel ICT-supported collaboration methodology leads to requirements documents of higher quality.*

High requirements quality, however, should not be gained by increased costs, i.e. spending more time. From the perspective of *analysts*, the new tool-supported collaborative methodology helps avoiding the many one-to-one interviews that usually occur during the elicitation process, which are particularly laborious if face-to-face communication is difficult to achieve or not possible at all. Moreover, due to the decentralized and rationalized requirements negotiation and selection procedure, the analysts should be mostly relieved from long discussions and processes of conflict resolution with users. Finally, the immediate feedback from the users during the specification phase should allow for faster responses by the analysts and hence lead to shorter cycle times and thus less effort for specification and validation (H2a in Figure 6).

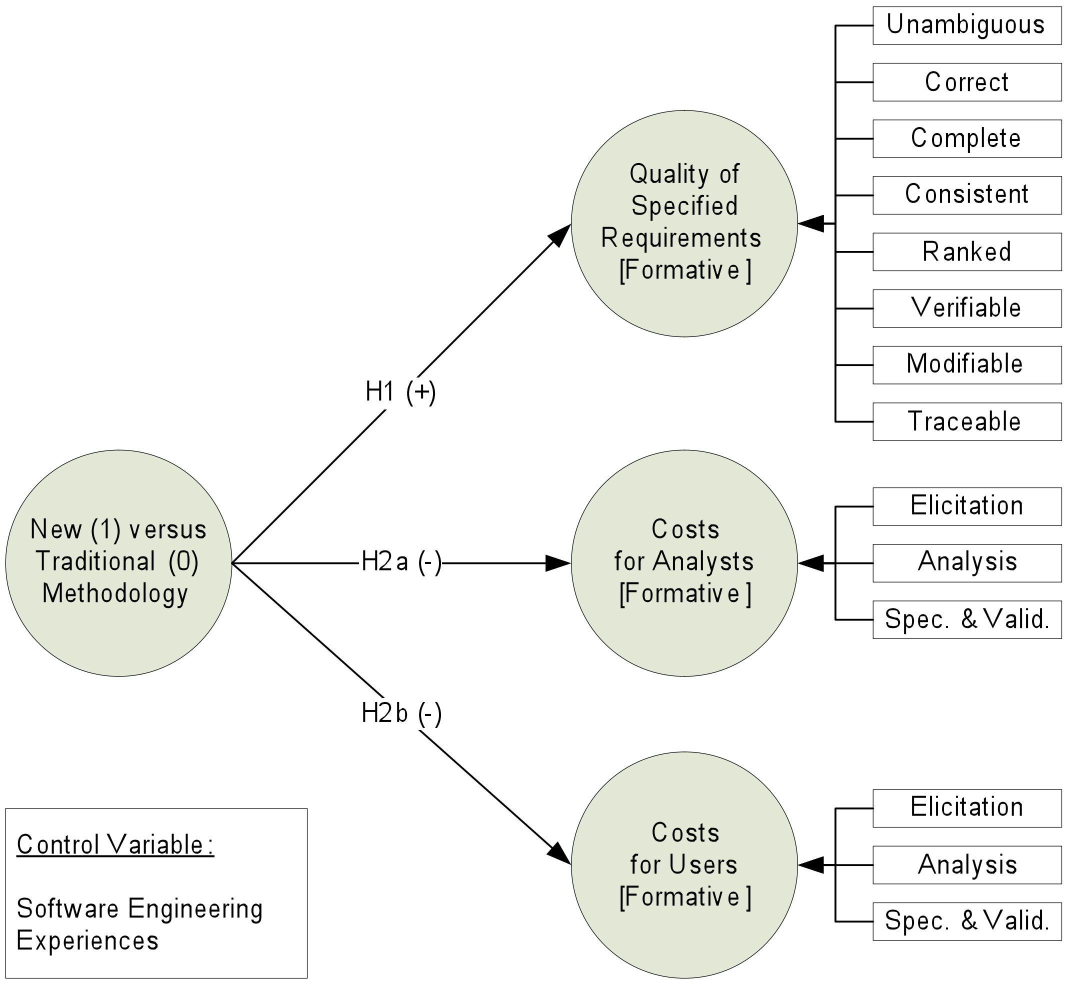


Figure 6 Model for Evaluating the Effect of the New Methodology

From the *users'* point of view, the avoidance of lengthy one-to-one interviews with analysts may also represent an efficiency advantage. Users may also benefit from immediately seeing the requirements of other users, thus avoiding redundancy in the first place while editing and changing requirements. The tool-supported assignment of values to requirements by means of the AHP may take some extra time but should pay off in terms of lower ensuing effort for requirements negotiation and selection. Finally, the validation of requirements should also take less time because misunderstandings may already have been smoothed out in earlier phases due to higher levels of collaboration, perspective taking, and mutual awareness of various stakeholders. Moreover, the new tool should make it easier to draw on previous descriptions that are stored in the requirements tracker and that can be retrieved accordingly for clarifications (H2b in Figure 6). Aggregating analyst and user effects, the following hypothesis is put forth:

Hypothesis 2a/b: *Compared to the traditional interview- and document-based methodology, the novel ICT-supported collaboration methodology is more cost efficient for both (a) analysts and (b) users.*

An overview of these hypotheses is graphically depicted in Figure 6. Each of the three dependent variables is an aggregate of a number of sub-dimensions. That is, overall quality is formed by the eight IEEE metrics (1998) and the costs for both, users and analysts, are broken down into the three sub-stages: elicitation, analysis, as well as specification and validation. Thus, each latent dependent variable is modelled in the so called *formative mode*, where each of the subdimensions (i.e. indicators) are conceptually independent making up a portion of the focal concept (Fornell 1989, p.161).

4.2 Experimental Settings and Subjects

For evaluating the effectiveness and efficiency of the novel ICT-supported methodology according to the hypotheses in Figure 6, a controlled experiment with student subjects within the scope of a programming term project was chosen (Hevner et al. 2004, p. 85). In this term project, the analysts were represented by 17 teams of *five to six students, i.e. software analysts* (SA), while the users were represented by *two* research assistants (RA). Within each of the student teams, one additional older and more experienced student tutor took on the role of the project manager (PM). All students had a technological background, studying either information systems or computer science. The term project was a mandatory part of their curricula, while the tutors (i.e. PMs) were hired as student assistants.

After conducting a technology-specific examination concerning Java and object-oriented programming prior to the experiment (with an ensuing initial drop-out rate of 22.52%), the starting sample consisted of 86 students (not including the PMs). Within this initial sample, the average student was 22.8 years old and in her/his third year of studies with a medium exposure in programming and SE projects. The actual project teams were composed according to the results of the Java exam and the experience questionnaire, while treatments are allocated randomly by means of a simple PHP script which automatically levels the teams with respect to their average software engineering skills. Hence, the groups had no past history of working together. *Nine* teams were allocated to the experimental treatment group (novel approach) and *eight* for control purposes (traditional approach). As can be inferred from Table 2, the average skill level of the experimental teams and the control teams was almost equal, with a score of 49.02 and 50.63 respectively (i.e. there is no statistically significant mean difference between the two groups, see Table 2). Overall, students correspond closely to young IT professionals with moderate to medium experience and can thus be expected to produce comparable results (Berander 2004; Hoest et al. 2000). In fact, most experimental evaluations have been conducted with student subjects (Sjoberg et al. 2005). The use of students is not only convenient in terms of cost efficiency (Fenton et al. 1994), but also in terms of performance, since students were often found to perform equally to professionals, in particular if the experiment took place as part of a term project which ensures the students' high commitment to the project (Berander 2004; Carver et al. 2003). Student commitment was ensured through the credits provided for the term project and by handing out a project development certificate by Intland Corp. in case of successful project participation (Carver et al. 2003).

4.3 Experimental Task

The experimental task was to specify the requirements for a 3D-version of a game known as "Dynablaster". The main aim of this game is to dexterously place time fuse ignited dynamite in a mazelike playing field and, thus, taking out opposing players and obstacles while also collecting extra credits for picking up bonus items, so-called "Power-Ups". Thus, the objective was to animate players, items, and explosions in a 3D manner. The Dynablaster project has been chosen for different reasons: (a) the 3D implementation requires innovative technology (Java 3D) and covers functionalities that also play an important role in business applications, such as appropriate user interface design, network interoperability, client-server architecture, and artificial intelligence, (b) "game programming [is] one of the most motivating topics for […] students" (Jimenez-Peris et al. 1999, p. 253), thus enhancing intrinsic motivation of the students (Damian et al. 2003b).

4.4 Experimental Procedure

The experiment was carried out over a period of six weeks encompassing the three distinct phases of requirements determination as introduced earlier. After a brief introductory period with additional lectures on RE practices in SE and a personal meeting of the groups (week 1), the teams had to interact remotely with the customers for *eliciting* an initial set of requirements (week 2), which had to be analyzed and consolidated in the course of the project. To ensure equal treatment of the groups by the customers in this initial phase, a standardized screenplay for the customers pertaining to both treatments was developed. This set of procedural guidelines contained (a) defined customer roles, (b) the information to be disclosed by each customer, and (c) different customer priorities, so that the analysts had to consolidate different views. The users also deliberately communicated superfluous and conflicting requirements in a standardized manner.

As a part of the *analysis* phase, particular requirements had to be selected to be either mandatory or optional for later implementation, while interdependencies as well as development costs and customer value needed to be determined according to the two treatments (weeks 3 and 4). Based on the analysis results, the requirements had to be *specified* according to the VOLERE standard (week 5) and reiterated with the customers for *validation* purposes (week 6). Each team had to develop and submit a validated set of requirements, either as one document (treatment 1) or through the requirements tracker (treatment 2, see Table 2). The sample solution was then taken as a basis for evaluating quality (for exact measurement procedures cf. section 4.5.1).

To support overall comparability and reproducibility, both, a set of standardized methodological and technological *controls* were specified for the development environment (Kitchenham et al. 2002, p. 727): First, in terms of development *methodology*, the teams were instructed to follow an agile and iterative approach where each student within one team was responsible for one particular set of requirements from elicitation to implementation and testing. Second, regarding *overall technology* support, codeBeamer was utilized as a standardized project management tool, integrating e.g. document and source code management (Robbins 2005). For both, procedural guidelines and tool support, additional lectures accompanying the term project were provided and standardized information material was handed out to the subjects in order to further level the overall skill level at a fair amount of experience.

4.5 Measurement of Experiment Variables

In order to measure requirements quality and achieve a high degree of comparability, objectivity, and thus validity, a *sample requirements document* was developed. The *Specification quality* of the teams' outcomes compared to the sample solution is determined by means of operationalizing the widely accepted "Practices for Software Requirements Specifications" (IEEE 1998)[8]

---

[8] IEEE standard 830-1998 is commonly used for requirements evaluation in both, academic literature and software development practice Gibson, C.B., and Abstract, J.L.G. "Unpacking the Concept of Virtuality: The Effects of Geographic Dispersion, Electronic Dependence, Dynamic Structure, and National Diversity on Team Innovation," *Administrative Science Quarterly* (51:3) 2006, pp 451-495, Smith, S., Lai, L., and Khedri, R. "Requirements Analysis for Engineering Computation: A Systematic Approach for Improving Reliability " *Reliable Computing* (13:1) 2006, pp 83-107, Toval, A., Nicolas, J., Moros, B., and Garcıa, F. "Requirements

4.5.1. Software Requirements Specification Quality

Correctness. Since "an SRS [software requirements specification] is correct if, and only if, every requirement stated therein is one that the software shall meet" (IEEE 1998, p. 4), the teams' submissions are compared to the sample solution by formal inspection.[9] Therefore, correctness is measured as percentage of correct requirements within the specification submitted, i.e. number of correct requirements divided by the number of overall requirements specified (see also the measure of precision, Baeza-Yates et al. 2002). For example, team 4 submitted 23 requirements, out of which 3 were superfluous and thus incorrect, the particular correctness score is 87%.

Unambiguity. Possibly ambiguous interpretations were identified manually by means of a formal inspection of the *correct* requirements with respect to (a) the terminology used and (b) the meaning conveyed by natural language. Then, the overall percentage of unambiguous requirements was determined by adding the percentage of unambiguous requirements in terms of wording and in terms of meaning and dividing this sum by two. For instance, team 11 specified 21 out of 22 (95%) correct requirements unambiguously in terms of terminology and 19 out of 22 (86%) with respect to the underlying meaning. Therefore, this team exhibits an overall unambiguousness score of 91%.

Completeness. A complete software requirements specification has to (a) include "all significant requirements", (b) define "responses of the software to all realizable classes of input", and (c) fully encompass "labels and references" (cf. IEEE 1998, pp. 5). Therefore, completeness is measured as the mean average of the percentages of *correct* requirements actually elicited[10], percentage of requirements with specified reactions to system inputs as

---

Reuse for Improving Information Systems Security: A Practitioner's Approach," *Requirements Engineering* (6:4) 2002, pp 205-219..

9 According to IEEE 1998, "there is no [automated] tool or procedure" to determine correctness (p. 4).

10 Thus, completeness is a measure of *recall* with respect to information retrieval performance figures Baeza-Yates, R.A., and Ribeiro-Neto, B.A. *Modern Information Retrieval* ACM Press / Addison-Wesley, 2002.. In

well as percentage of requirements correctly labeled and referenced within the software requirements specification glossary with respect to the sample solution. For example, team 4 elicited 20 out 23 (87%) correct requirements, specified reactions to system inputs for 15 of those (75%), and fully referenced all correct requirements (100%). The mean average is therefore 87%.

Consistency. According to IEEE 1998, this software requirements specification quality measure refers to *internal* consistency and is determined by manual inspection as the percentage of requirements that are not involved in any conflicts with other requirements, i.e. team 9 has specified 18 out 19 correct requirements without inherent conflicts and thus scores a consistency figure of 95%.

Ranking. Within this experiment, "ranking for importance and/or stability" (IEEE 1998, pp. 6) is measured by manually checking whether prioritization information for particular requirements is given, since objectively evaluating the priorities' adequacy is impossible. Hence, ranking is measured as the percentage of correctly prioritized requirements.

Verifiability. With regard to IEEE 1998, requirements are measured as verifiable by manual inspection if their manner of specification allows for checking whether the final software product (i.e., Dynablaster 3D) meets this particular requirement or not. As part of this measurement procedure, ambiguous requirements are automatically marked as not verifiable (cp. unambiguity). Therefore, verifiability is defined as percentage of verifiable requirements out of the correctly specified ones. Team 9, for instance scored 19 correctly specified requirements, out of which 17 are verifiable; thus verifiability amounts to 89%.

Modifiability. In order to be modifiable, a specified software requirement needs to (a) exhibit a clear and coherent organization, (b) not be redundant, and (c) "express each

---

contrast to correctness, *missing* requirements are considered, while correctness (or precision) penalizes superfluous ones.

requirement separately" (IEEE 1998, p. 8). Hence, modifiability is measured as mean average of these components, whereby *organization* is evaluated with respect to the percentile completeness of the table of contents and the use of explicit cross-references.

Redundancy is measured as percentage of *correct* requirements without redundant contents and atomicity as percentage of requirements that cannot be subdivided into two or more meaningful requirements. As an example, team 17 was missing 2 cross-references pertaining to 19 correctly specified requirements (89%), out of which 2 were also redundant in contents (89%), while 5 requirements were further decomposable, i.e. atomicity is 74%. Thus, modifiability for team 17 is 84%.

Traceability. A software requirements specification is traceable if (a) the *origin* of each requirement is clear (backward or source traceability) and (b) it facilitates referencing of particular requirements in ensuing project phases (forward or design traceability; IEEE 1998 and Sommerville (2004)). Therefore, "bi-directionality" is to be achieved (cf. CMMI standard, SEI 2002). *Backward traceability* is checked by formal inspection and measured as percentage of *correct* requirements. These requirements should be attributed to their origins in terms of customers and respective documents with a weight of 80% of the overall traceability figure, whereas *forward traceability* only accounts for unique requirements identifiers weighted with 20%.[11] Overall traceability thus is determined as weighted average of backward and forward traceability.

4.5.2. Cost-Efficiency of Requirements Determination

Cost-efficiency was measured in terms of man-hours in each project phase (i.e. elicitation, analysis, and specification/validation) for both analysts and users. The overall

---

[11] Backward traceability requires rigorous documentation during requirements determination and is hardly reproducible ex post, whereas forward referencing can be added later Gotel, O.F., A. "An Analysis of the Requirements Traceability Problem," Proceedings of the 1st International Conference on Requirements Engineering (ICRE'94), IEEE Computer Society Press, 1994, pp. 94-101.. Therefore the weighting is shifted on a scale of four to one in backward direction (IEEE 1998).

effort of analysts was determined in two different ways. While the effort of the project managers was assessed based on a weekly survey, the analysts, i.e. students, recorded their effort self-dependently.

Requirements elicitation costs comprise the analyst and customer effort spent for all activities leading to a first set of requirements, e.g. the initial meeting, requirements identification, and formulation, whereas *requirements analysis costs* measure the man-hours spent by both parties for determining interdependencies, prioritizing, cost and value estimation as well as requirements selection procedures. Eventually, *requirements specification and validation costs* capture the effort for structuring the formal specifications and commenting on the particular requirements' validity, respectively.

5 HYPOTHESES TESTING

For hypotheses testing two types of analyses were performed. First, the means of the efficiency and effectiveness measures for both the control groups (using the traditional methodology) and the treatment groups (using the novel methodology) were compared. For this purpose, both the absolute difference between the mean values and the level of statistical significance (via T-test) of that difference were calculated. Second, a partial least squares (PLS) analysis (Chin 1998) was performed for assessing the relative impact of the chosen methodology (traditional versus new) on user and analysts efficiency as well as effectiveness. PLS was chosen, first, because it allows to model latent variable models with formative indicators. As shown in Figure 6, both efficiency and effectiveness were treated as latent variables that are formed by the respective sub-dimensions, i.e. measures. Second, PLS makes less demand on sample size than alternative structural equation modeling procedures and it does not require normally distributed data (Chin 1998). Notably, our sample size of 17 groups is rather low and normal distribution was violated for some variables (see Table 2, 3).

With PLS, the strength of a relationship is indicated by the path coefficients between the independent variables (type of methodology) and the respective dependent variables (effectiveness, costs for users, and costs for analysis). Theses path coefficients can be interpreted similarly to regression coefficients. Tests of significance were obtained using the bootstrap routine (Chin 1998). In addition, the PLS analysis provides the indicator weights that express the strength with which each indicator forms a given construct (Chin, 1998c). In this study, the combination of the path coefficients and the weights provides information about which effectiveness and efficiency measures are most affected by the new methodology and in which direction (positive or negative).

5.1 Impact of New Methodology on Effectiveness

Table 2 shows the mean differences of the effectiveness measures between the control group and the treatment group. The strongest differences exist in the completeness, modifiability, and verifiability of the software requirements. The groups using the new methodology perform significantly better regarding completeness ($p<0.001$) and modifiability $p<0.01$). In contrast, the verifiability of requirements is significantly better in the control group (although it should be noted that for this variable, the data was not normally distributed based on checks of skewness and kurtosis (Mardia 1970)). The other factors, including the control variable software engineering knowledge, are not significantly different between groups. Notably, there was absolutely no difference in ranking ability and traceability between the groups. The scores for these metrics were 100% for all groups.

Table 2 Results of Mean Comparison for Effectiveness

	SE knowledge	Correct	Unambiguous	Complete	Consistent	Ranked
Mean Control Group	50.625	0.963	0.959	0.851	0.981	1.000
Mean Treatment Group	49.019	1.000	0.958	0.926	0.995	1.000
Diff. Control-Treatment	1.606	-0.037	0.001	-0.075	-0.013	0.000
Normally Distributed?	YES	NO	YES	YES	NO	NO
p-Value	0.167	0.174	0.929	0.002	0.228	n/a
Significant Difference?	NO	NO	NO	YES	NO	---
Level of Significance	---	---	---	p<0.001	---	---

	Verifiable	Modifiable	Traceable
Mean Control Group	0.925	0.859	1.000
Mean Treatment Group	0.879	0.921	1.000
Diff. Control-Treatment	0.046	-0.061	0.000
Normally Distributed?	NO	YES	NO
p-Value	0.045	0.007	n/a
Significant Difference?	YES	YES	---
Level of Significance	p<0.05	p<0.01	---

Similar results are obtained from the PLS analysis (see Figure 7). First, there is a significant positive effect of using the new methodology on effectiveness, i.e. the overall quality of software requirements (β=0.798, t=2.91, $p<0.01$). Taking a closer look at the quality indicators reveals that the *completeness* of requirements has the highest weight (w) and is the only indicator that is statistically significant (w=0.562, t=1.24, $p<0.10$). In line with the results from the mean differences, *modifiability* has the second largest weight, albeit not significant (w=0.466, t=0.70). It is also notable that verifiability and correctness have negative weights.

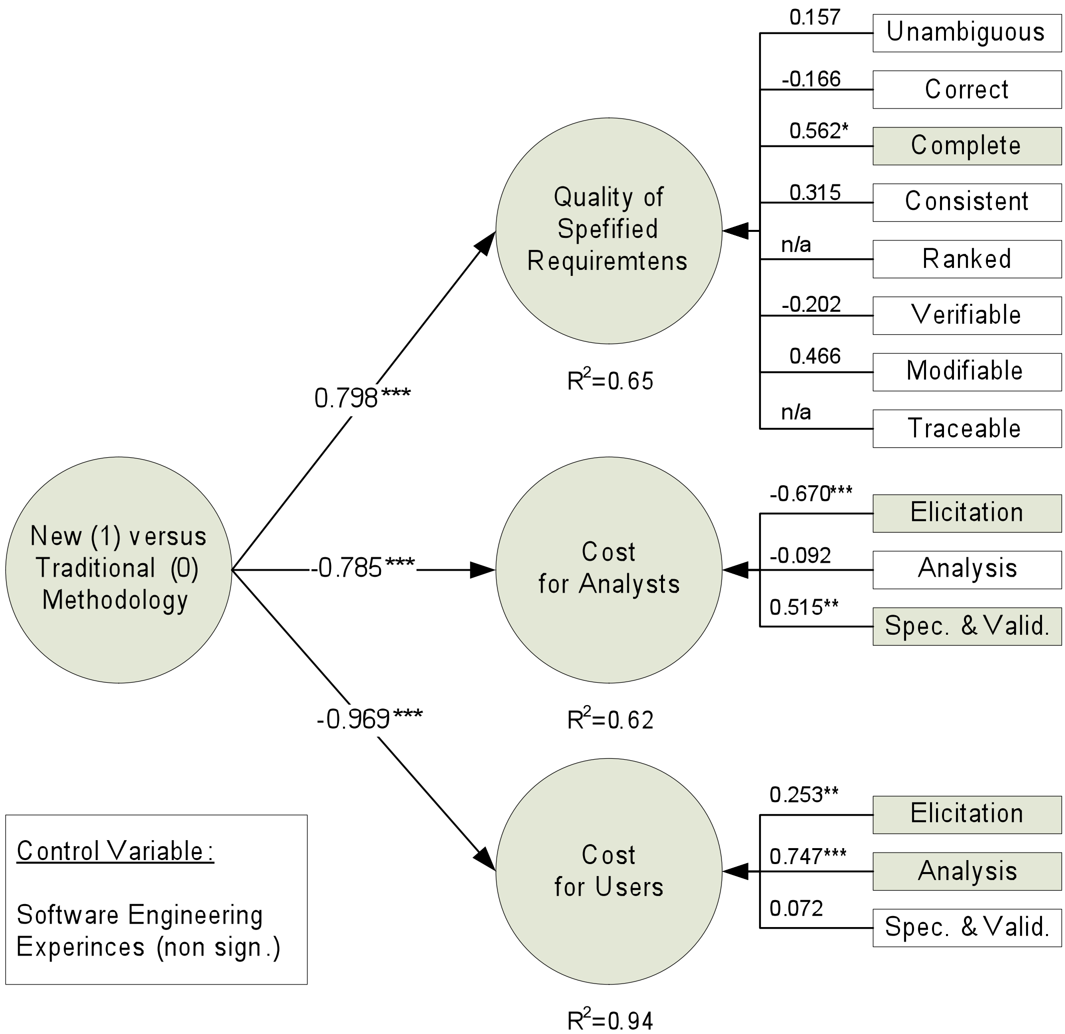


Figure 7 PLS Results of Hypotheses Testing

5.2 Impact of New Methodology on Efficiency

The mean differences in efficiency are presented in Table 3. As can be inferred form Table 3, the groups using the new methodology on average show lower costs for requirements elicitation, analysis, and specification & validation than the control group using the traditional procedure. In particular, the effort for requirement analysis is significantly reduced (Δ mean = 1.581, p<0.001). The effect on the costs for analysts are less consistent. While the costs of specification & validation are most strongly and significantly reduced (Δ mean = 6.888, p<0.01), the costs for elicitation significantly increase (Δ mean = -4.120, p<0.01), with analysis costs showing no difference at all.

Table 3 Results of Mean Comparison for Efficiency

	Costs for Users			Costs for Analysts		
	Elic.	**Analy.**	**SpecVal.**	**Elic.**	**Analy.**	**SpecVal.**
Mean Control Group	2.020	2.375	0.355	6.500	6.625	9.688
Mean Treatment Group	1.586	0.794	0.249	10.620	5.448	2.800
Diff. Control-Treatment	0.434	1.581	0.106	-4.120	1.177	6.888
Normally Distributed?	YES	YES	YES	YES	YES	NO
p-Value	0.000	0.000	0.011	0.002	0.252	0.007
Significant Difference?	YES	YES	YES	YES	NO	YES
Level of Significance	p<0.001	p<0.001	p<0.05	p<0.01	---	p<0.01

This result is completely in line with the PLS analysis. As can be inferred from Figure 7, both the costs for users (β= -0.969, t=86.8, $p<0.001$) and the costs for analysts (β= -0.785, t=9.0, $p<0.001$) are decreasing with the usage of the new methodology, as reflected by the strong negative and significant path coefficients. By closer examination of the weighting scheme, the costs for the users are mostly reduced due to lower costs of requirements analysis (w=0.747, t= 6.03, $p<0.001$). By contrast, the costs for analysts are reduced for the specification & validation (w=0.515, t=1.45, $p<0.1$), while they increased for the elicitation (w= -0.670, t=2.25, $p<0.05$).

6 DISCUSSION

This study has contributed to a better understanding of the design of an ICT-supported methodology for distributed requirements determination by being the first to explicitly use a design theory for constructing a novel methodology and a tool as artifact and justifying hypotheses about improvements in efficiency and effectiveness compared to an alternative methodology. It is one of the first studies that evaluated a newly designed methodology within an experiment that provides important information about the usefulness and limitations of the new methodology in a controlled setting.

By using Te'eni's (2001) cognitive-affective model of organizational communication for IT design as a conceptual basis, the choice of functionalities of the new methodology has been theoretically grounded. The theoretical foundation of the design allows for a meaningful interpretation of the experimental results by drawing on the design elements of Te`eni's theory. Although the experiment may not be viewed as a complete empirical test of Te'eni's design theory (Majchrzak et al. 2005a) since none of its propositions were explicitly tested (Hannay et al. 2007), the evaluation of the resulting prototype still provides one important step forward in the process of testing a design theory (Walls et al. 1992). Thereby, the experiment, on the one hand, provides "…essential feedback [on] the quality of the design process" [i.e., Te`eni's design theory and the way it has been applied] (Hevner et al. 2004, p. 85). On the other hand, however, the experiment also provides important hints at possible improvements of "…the design product under development" [i.e., the new ICT-supported methodology] (Hevner et al. 2004, p. 85). Accordingly, both implications for the design as well as for theory are presented.

6.1 Implications for Design

The objective of the new ICT-supported methodology was to alleviate the risk of increasing communication costs and decreasing communication effectiveness of requirements determination in distributed environments. The experiment provides a first test of the appropriateness of the new methodology for reaching these objectives by comparing the efficiency and effectiveness of two sets of user-analysts groups: one using a traditional form of communication via synchronous verbal communication as well as electronic document exchange and the other using the newly designed internet-based platform with collaboration and decision support functionality.

By closer examination of the experiment results, it is particularly intriguing to see the varying effects that the new methodology has on specific measures of effectiveness and efficiency.

Effectiveness, i.e. quality improvements can mostly be ascribed to higher levels of completeness and modifiability of the final requirements documents. The higher *completeness* can be explained by the wiki-based collaboration platform. The co-creation of requirements documents during the elicitation phase allows users and analysts to learn from each other and to complement each others' base of knowledge (Majchrzak et al. 2008). This results in a more detailed description of requirements as compared to the one-to-one interview-based procedure, where users and analysts are constrained to their own domain knowledge. Thus, the decentralized mechanism for knowledge utilization through the wiki is more effective than the more centralized procedure (Majchrzak et al. 2008), where the analysts seek to extract the relevant knowledge from single users. Obviously, the wiki helps to better utilize the distributed knowledge of users and analysts – a central problem of economic society at large:

> "The economic problem of society is …a problem of the utilization of knowledge which is not given to anyone in its totality" (Hayek 1945, p.77-78)

The more detailed description of requirements and the consideration of multiple views by users and analysts, however, may cause the drawback of increased information overload (Te'eni 2001). For that reason, it is particularly pleasing to see that *modifiability* has also improved through using the new collaborative methodology. This improvement may be attributed to tool support for highlighting dependencies among the group of analysts as well as supporting requirements tracing from their origin to their final state. Thus, higher contextualizing and perspective taking come along with support for document organization and it is this joint communication strategy which may explain why both completeness and modifiability were better off with the new methodology. Interestingly, *verifiability* tended to be higher with the traditional approach. However, verifiability is also the one metric where user involvement may be least necessary. In order to ensure verifiability, it is important to link user requirements with specific implementation issues. This requires profound implementation knowledge which is the domain of the analysts. Thus, the personal verbal meetings among analysts in the traditional approach may have been advantageous for ensuring verifiability (Guinan et al. 1998). This interpretation substantiates the quest for the use of mixed media in requirements determination where the wiki-enabled elicitation allows for achieving common ground among users and analysts (i.e., common knowledge and the awareness of it (Clark et al. 1991)), while synchronous verbal communication (e.g., through video conferencing) may be needed as a complementary communication medium for the analysts in the subsequent analysis and refinement of requirements (Damian et al. 2008).

The results of the *efficiency* evaluation also provide a differentiated picture. The users clearly have the largest benefit during the *analysis* phase. Obviously, the tool support via AHP makes it easier for users to determine the relative utility of requirements. In addition, the graphical representation of the cost-value ratios increases transparency and understandability of requirements selection which helps avoiding lengthy discussions about relative advantages of requirements. The *elicitation* is also significantly alleviated, albeit less than the subsequent analysis. This may again be attributed to the co-creation of requirements which reduces the redundant editing of requirements among the users and also allows for quick feedback from the analysts. This again reduces the need for clarifying dependencies later on and helps avoiding misunderstandings. From the *perspective of analysts*, however, the strong demand for collaboration during requirements elicitation significantly increases their effort compared to the traditional interview-based procedure. Apparently, the analysts increase their up-front investments into structuring requirements, providing feedback, and co-editing requirements when the possibility for jointly creating and incrementally enhancing the documents with the users is given. Thus, tool support for contextualization and perspective taking actually encourages analysts to invest more into the elicitation phase. Notably, however, this up-front investment pays off by significantly decreasing the effort for the specification and validation of requirements. Most of the iterative controlling and testing already takes place during elicitation.

Altogether, the design implications can be summarized as follows. The wiki-functionalities should come along with functional support for organizing and selecting requirements, which substantiates the chosen design of our new tool-supported methodology. There is slight evidence, however, that complementary support for synchronous verbal communication among analysts may be fruitful in the analysis phase to ensure the verifiability of requirements. Alternatively, some innovative tool support for enhancing verifiability may be thought of.

6.2 Theoretical Implication

Based on the preceding discussion two theoretical implications may be derived. First, this study has substantiated the quest for a dynamic perspective when designing ICT for supporting communication (Te'eni 2001). In the early stages of communication, it is important to establish common ground among users and analysts (Damian et al. 2008). Tool support for high contextualization and perspective taking has been shown to be particularly effective for this purpose. However, as communication is purposeful for reaching a common outcome (e.g., mutually agreed requirement specifications), it becomes necessary to move from contextualization towards decontextualization and abstraction (Westrup 1999) in order to be able to make effective decisions (e.g. selecting requirements and reaching agreement on a final set of requirements). For this purpose, ICT functionality that supports organizing and formalizing information plays a more prominent role. The decontextualization, however, should not occur at the expense of loosing detailed descriptions. Indeed, for managing change requests later on (i.e., ensuring modifiability) it is essential that detailed context information and the historical paths of communication can be traced. ICT-supported collaboration platforms have the potential to overcome the trade-off between contextualization and decontextualization (Westrup 1999) by dynamically linking the former with the latter.

This dynamic view of communication and associated ICT support is closely related to the second implication. As noted by Te'eni (2001, p. 294): "Decomposing the communication process into sub-processes brings closer the possibility of developing more specific design guidelines for such systems." In this study, it has been proven beneficial to break down the requirements determination process into three sub stages and to construct design features for each stage separately. However, it has also been shown that it is important to consider the interdependency between the sub-stages as well as the transformation of work practices that ICT support for particular sub-stages brings along. Enabling distributed collaborative work through a wiki environment for requirements elicitation has significantly changed the way in which users and analysts are interacting as opposed to the traditional procedure of synchronous verbal communication. For example, analysts took much more effort in the elicitation phase for the benefit of lower effort in later stages. This is illustrated by Figure 8 where the impact of investments in earlier stages on subsequent stages is analyzed through a PLS analysis. As can be inferred from Figure 3, the elicitation effort for those analysts that used the novel ICT-supported methodology led to significantly lower effort in the analysis (β= -0.406, $p<0.1$) and the specification & validation phase (β= -0.324, $p<0.1$). By contrast, higher effort in the elicitation only led to significantly less effort for requirements analysis (β= -0.253, $p<0.01$) and there are even indications for adverse effects for the other two paths (albeit non significant). Thus, the use of the new ICT-supported methodology transforms work practices (Orlikowski 1996) and partially changes the causal relationships between the sub-processes of requirements determination.

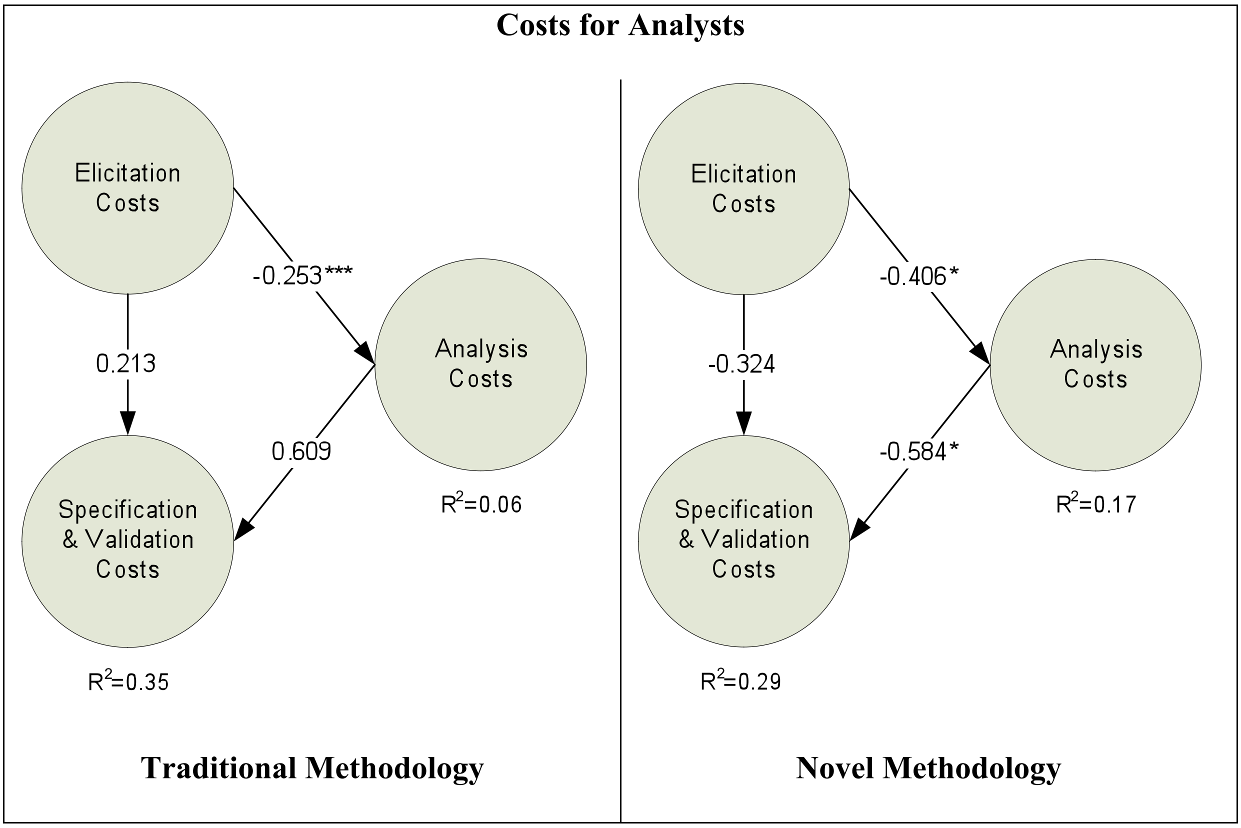


Figure 8. Comparison of Dynamic Impacts between Traditional and New Methodology

6.3 Limitations

In order to put the previously discussed results into perspective, a number of possible limitations to this research should be considered. With respect to its *internal* validity, a mono-operations bias can be criticized, since only one experimental task (i.e., the determination of requirements for a computer game) was considered. The observed efficiency and effectiveness impacts may likely vary for other software development projects. For example, the impact of ICT-support for contextualization on collaboration know-how development was shown to be stronger with non-routine tasks rather than routine task (Majchrzak et al. 2005b). Thus, for development projects that are routine or less asset specific (Dibbern et al. 2008) than the one studied in this experiment, the effect of the ICT-supported methodology may likely vary as well. In addition, the experimental project involved a limited number of requirements and a limited number of stakeholders (users and analysts). Therefore, it is not clear whether the experimental results discussed in the previous sections can get generalized for more complex projects.

Moreover, *process conformance* regarding the experimental procedure and tool usage can also be questioned. As means of procedural enforcement and environmental control, both the projects' trackers and log files were checked and the tutors collected anonymous feedback regarding procedural correctness from their respective teams.

The limited *external validity* of the experiment due to student subjects may still be a factor, although (a) motivation and commitment of the subjects was ensured through incentives (Berander 2004) and (b) graduate students with advanced software engineering skills were selected that perform comparable to young IT professionals (Hoest et al. 2000).

Furthermore, with respect to generalizability, the agile and collaborative requirements determination methodology designed here as well as the best practices taken as benchmark for evaluation cannot be applied to all industrial settings, such as development of dependable software. From a theoretical design perspective, on the other hand, Te'eni's (2001) design theory yet awaits a *complete* empirical test, although Te'eni recommended utilizing it for guiding ICT design.

7 SUMMARY AND CONCLUSION

This article intends to overcome the problem of communication barriers and deficiencies between locally dispersed users and analysts in the process of requirements determination. For this purpose, a novel ICT-supported methodology for distributed requirement determination has been designed which supports the elicitation and analysis, the specification and the validation of information systems requirements. The key design features of the corresponding instantiated ICT based tool have been developed by utilizing a acknowledged design theory, exemplified by Te'eni's (2001) cognitive-affective model of organizational communication for designing IT. This study attempts to develop technology grounded on theory. The result is a novel ICT-supported methodology that enables rich contextualization and perspective taking of various stakeholders during requirements elicitation as well as support for organization, formalization, and aggregation of information during requirements analysis and specification. Thus, one of the unique features of this study is that it supports different requirements determination processes with different ICT-features on a subtask level which enables a better task-technology fit (Zigurs et al. 1998). The new tool integrates collaboration technology (i.e., a wiki-based system) with templates for structuring information, as well as with decision support functionalities including graphical representations. Thereby, a platform for achieving common ground (Clark et al. 1991) between users and analysts has been achieved.

Furthermore, the novel approach has been carefully evaluated during an experiment with 17 distributed user/analyst groups that were required to specify requirements for a new software product over a period of six weeks. The evaluation showed that, on average, the nine groups that used the collaboration platform outperformed the eight groups that used a traditional procedure of verbal synchronous communication and electronic document-exchange. The positive effects of the new methodology on both efficiency and effectiveness were further confirmed through a multivariate analysis through PLS, where the impacts of variations in the type of methodology (novel versus traditional) on both efficiency and effectiveness metrics were tested. Both the examination of mean differences and the multivariate analysis revealed overall positive effects, however, they also highlighted dynamic effects of the new concept. It could be observed that the new tool led to higher effort for the analysts during the elicitation phase as compared to the traditional procedure. However, these higher “up-front investments” into achieving common ground with the users paid off through significantly lower effort for the subsequent analysis as well as specification and validation of requirements. Thus, this study also contributes to a better understanding of the dynamics in the process of requirements determination and it highlights the effects of ICT-support for knowledge accumulation of interrelated tasks that build on each other.

These findings have important implications for future research on the effect of ICT on organizational processes. Research should take a long term perspective and consider path dependencies of ICT support for consecutive and recurring stages of communication and interaction between users and clients. Future research may also seek to explicitly consider the effects of collaborative ICT tools on (multiple) organizational boundaries. For example, the question is raised whether the use of collaborative ICT tools may also help to overcome organizational boundaries due to different (culturally grounded) values and norms between client and vendor personnel (Te'eni 2001). Moreover, interaction effects between task properties, like high task specificity, and cognitive distance, exemplified by differing values and norms (Dibbern et al. 2008), with the effect of ICT supported collaboration should be carefully studied. Thus, the question is raised whether the effect of ICT support on the efficiency and effectiveness of collaboration in requirements engineering is contingent on the type of task or business process that is being supported by the software product or whether it depends on the relational attributes between users and analysts (such as the level of mutual trust). Studying such effects would allow getting additional insights into potential constraints or necessary extensions in the current functionality of our collaborative ICT tool.

APPENDIX A

Table 1: Citation Analysis of Existing Requirements Determination Approaches

Requirements Engineering Method	**Source**	**Number of Citations**
AGORA	Kaiya et al. (2002)	63
ARENA	Gruenbacher and Braunsberger (2003)	15
Cost-Value Approach (CVA)	Karlsson and Ryan (1997)	235
EGRET	Sinha et al. (2006)	11
Easy WinWin (EWW)	Boehm et al. (2001)	144
KAOS	van Lamsweerde et al. (1998)	306
RM-Tool	Lang and Duggan (2001)	27
PREView	Sommerville et al. (1998)	80
Scenario-Based Requirements Analysis (SBRA)	Sutcliffe (1998)	69
Volere	Robertson and Robertson (2006)	511

Table 1 contains a simple citation analysis based on the number of references the related work achieved as an indicator of impact and relevance.

Table 2a: Evaluation Framework for Requirements Determination Approaches

Evaluation Category	**Sub-Category**	**Sub-Sub-Category**
Scope	• Requirements Elicitation and Analysis	Elicitation (E); Classification (C); Negotiation (N); Prioritization (P); Selection (S)
	• Requirements Specification and Validation	Specification (Sp.); Validation (V)
Tool Support	• Provided • Continuity	n/a
Evaluation	• Applicability • Descriptive Advantages • Empirical Evaluation	n/a

Table 2a outlines the three main evaluation criteria applied to related requirements determination approaches. “Scope” refers to the coverage of a standardized set of requirements determination activities, while “tool support continuity” evaluates technological aspects.

Table 2b: Evaluation of Existing Approaches to Requirements Determination

Approach	**Scope**							**Tool Support**		**Evaluation Method**		
	Elicitation & Analysis					Sp.& V.						
	Elicitation	Classifaction	Negotiation	Prioritization	Selection	Specification	Validation	Provided	Continuity	Applicability	Descriptive	Empirical
AGORA	+	++	++	+	+	+	-	-	n/a	+	+	-
CVA	-	-	-	++	+	-	-	+	-	++	+	-
EGRET	++	+	++	-	+	+	-	+	++	++	+	+
EWW/ARENA	++	++	++	+	-	-	-	++	+	++	+	-
KAOS	+	++	-	-	-	+	-	++	+	++	+	-
PREView	++	++	++	-	-	-	-	-	n/a	++	+	-
RM-Tool	-	+	+	-	+	++	+	+	+	+	+	+
SBRA	++	+	-	-	-	+	++	+	+	+	+	-
Volere	++	++	++	+	+	++	+	++	+	++	+	-

Table 2b contains an overview of the evaluation of existing requirements determination approaches in terms of the criteria defined in table 2a.

Appendix B

Table 1: Attribute values of communication complexity elements

Communication complexity elements	**Proposed attribute value**	**Rationales / Remarks**
Dynamic complexity	High	Time pressures, task variety, and number of involved actors are complexity drivers
Cognitive complexity	High	Low analyzability, high variety and high temporality of requirements determination tasks lead to high interdependencies between actors and tasks, high multiplicity of tasks and abstractions as well as high incompatibility of task representations
Affective complexity	Medium	Varying professional backgrounds and IS designer values of actors which imply somewhat differing affective models, values and norms[12]

[12] During the design of the novel approach, intercultural aspects of distributed requirements determination have not been explicitly incorporated.

Table 2: Impact of communication complexity on communication strategy

Communication strategy elements	**Proposition in the Te'eni model and cause**	**Proposed attribute value by Te'eni**	**Realized attribute value in novel approach**	**Rationals / Remarks**
Contextualization	2A (for high cognitive complexity)	High	High	Important
Affectivity	2B (for high affective complexity)	High	Medium	Due to medium affective complexity
Control by testing and adjusting	2C (for high cognitive complexity coupled with high dynamic complexity)	High	High	Important
Control by planning	2D (for high cognitive complexity coupled with low dynamic complexity)	Low	Low	
Perspective taking	2E (for high cognitive complexity coupled with high affective complexity)	High	Medium	Due to medium affective complexity
Attention focusing	Not determined		High	

Table 3: Impact of communication inputs on communication strategy

Communication process elements	**Proposition in the Te'eni model and cause**	**Proposed attribute value by Te'eni**	**Realized attribute value in novel approach**	
Frequency of requesting information	8 A (due to high task variety)	Frequent	Frequent	
Contextualization	8 B (due to low task analyzability)	High	High	Important
Control of testing and adjusting	8 C (due to short time to complete the task)	High	High	Important
Control by planning	8 D (due to long time to complete the task)	Low	Low	Not relevant
Contextualization	9 A (due to greater cognitive distance)	High	High	Important
Frequency of requesting information	9 B (due to greater cognitive distance)	Frequent	Frequent	
	9 C (due to greater affective distance)	Infrequent	Frequent	Not relevant: affective distance is low
Frequency of communication	10 A (for interdependent cultures and limited affective and cognitive distance)	High	High	
Perspective taking	10 B (for interdependent cultures)	High	High	

Table 4: Impact of communication strategy on communication medium

Communication medium elements	Proposition in the Te'eni model and cause	Proposed attribute value by Te'eni	Realized attri-bute value in novel approach	Rationales / Remarks
Interactivity	3A (for high control – testing and adjusting)	High	High	Important
Channel capacity	3 B (for high contextualization)	High	Medium	Compensated by 4a due to limitations of asynchronous technologies
	3 C (for high affectivity)	High	Medium	Affectivity is considered medium in our design scenario
Adaptiveness	3 D (for high perspective taking)	High	Medium	Due to medium perspective taking

Table 5: Reverse impact of communication medium on communication strategy

Communication strategy elements	Proposition in the Te'eni model and cause	Realized attri-bute value in novel approach	Rationales / Remarks
Increasing control by testing and adjusting	4 A (for low channel capacity coupled with high interactivity)	High	Important
Increasing control by planning	4 B (for low channel capacity coupled with low interactivity)	Low	Not important: control by planning does not apply
Decreasing affectivity	4 C (for low channel capacity coupled with low interactivity	Low-Medium	Not important: affectivity is low
Increasing affectivity	4 D (for low channel capacity and high interactivity)	Low-Medium	

Table 6: Impact of communication strategy on message form

Communication form elements	Proposition in the Te'eni model and cause	Proposed attribute value by Te'eni	Realized attri-bute value in novel approach	Rationales / Remarks
Message distribution	5 A (for high affectivity)	Small	High	Due to low assumed affectivity
Message organization	5 B (for high contextualization)	High	Medium	In order to foster creativity and spontaneous communication
	5 C (for high control by planning)	High		Irrelevant since control by planning is low
Message formality	5 D (for high affectivity)	Low	Medium	Due to low affectivity
	5 E (for high control by testing)	High	High	Important for specification and validation
	5 F (for contextualization)	Low	Medium	Important for elicitation and analysis

Table 7: Reverse impact of message form on communication strategy

Communication strategy elements	Proposition in the Te'eni model and cause	Realized attri-bute value in novel approach	Rationales / Remarks
Increasing attention focusing	6 A (through long message size)	Short message size	Counter-intuitive: smaller message increase the attention of receivers
Increasing control through adjusting and testing	6 B (through low message organization; provided media interactivity is high)	Medium message organization	

Table 8: Impact between medium and message form

Communication form elements	Proposition in the Te'eni model and cause	Realized attri-bute value in novel approach	Rationales / Remarks
Short message size	7 A (through high interactivity)	Short message size	Important
High message formality	7 B (through low channel capacity)	Medium message formality	Due to medium channel capacity

Appendix C

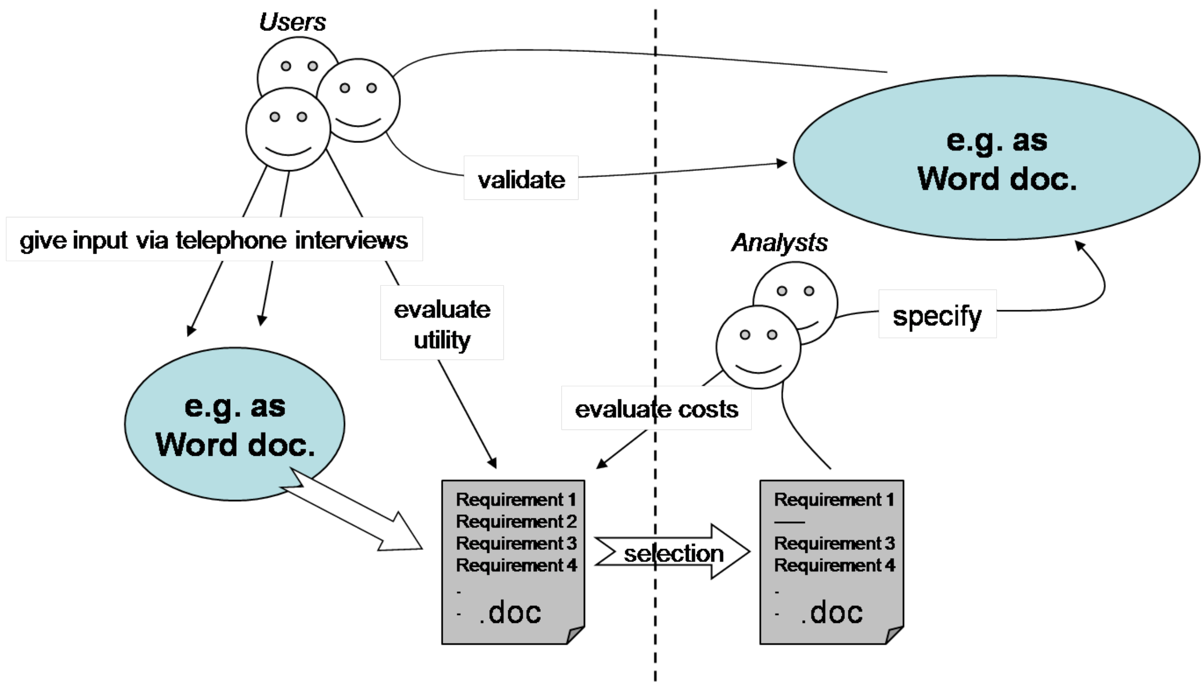


Figure 1: Process Overview of Alternative Requirements Determination Methodology

Appendix D

Table 1 Data for Control and Treatment Group

	No.	Group	SE knowledge	Correct	Unambiguous	Complete	Consistent	Ranked
Control Group (Traditional Methodology)	3	0	54.800	0.833	0.950	0.823	0.950	1.000
	4	0	52.600	0.870	0.975	0.873	1.000	1.000
	5	0	50.800	1.000	0.952	0.876	1.000	1.000
	6	0	51.000	1.000	0.975	0.790	1.000	1.000
	8	0	49.600	1.000	0.974	0.837	1.000	1.000
	9	0	50.000	1.000	0.947	0.837	0.947	1.000
	14	0	48.600	1.000	0.952	0.955	0.952	1.000
	17	0	47.600	1.000	0.947	0.819	1.000	1.000
Treatment Group (Novel Methodology)	1	1	50.000	1.000	0.967	0.957	1.000	1.000
	2	1	51.200	1.000	0.935	0.971	1.000	1.000
	7	1	51.170	1.000	0.977	0.910	1.000	1.000
	10	1	47.600	1.000	0.975	0.890	1.000	1.000
	11	1	44.400	1.000	0.909	0.910	1.000	1.000
	12	1	51.000	1.000	0.955	0.925	1.000	1.000
	13	1	47.200	1.000	0.955	0.895	1.000	1.000
	15	1	49.800	1.000	1.000	0.923	0.952	1.000
	16	1	48.800	1.000	0.952	0.955	1.000	1.000

Table 2 Data for Control and Treatment Group (2 from 3)

	No.	Group	Verifiable	Modifiable	Traceable	Elic_User	Analy_User	SpecVal_User
Control Group (Traditional Methodology)	3	0	0.950	0.767	1.000	1.830	2.270	0.380
	4	0	0.950	0.850	1.000	2.170	2.400	0.420
	5	0	0.905	0.889	1.000	2.330	2.420	0.370
	6	0	0.950	0.900	1.000	2.000	1.970	0.470
	8	0	0.895	0.825	1.000	2.000	1.800	0.270
	9	0	0.895	0.912	1.000	2.170	2.470	0.250
	14	0	0.905	0.889	1.000	1.830	2.970	0.330
	17	0	0.947	0.842	1.000	1.830	2.700	0.350
Treatment Group (Novel Methodology)	1	1	0.826	0.870	1.000	1.480	0.730	0.250
	2	1	0.870	0.928	1.000	1.580	0.930	0.400
	7	1	0.864	0.955	1.000	1.720	0.900	0.200
	10	1	0.950	0.933	1.000	1.680	0.730	0.200
	11	1	0.772	0.939	1.000	1.530	0.770	0.180
	12	1	0.909	0.879	1.000	1.520	0.830	0.180
	13	1	0.909	0.894	1.000	1.480	0.700	0.250
	15	1	0.905	0.937	1.000	1.550	0.830	0.350
	16	1	0.905	0.952	1.000	1.730	0.730	0.230

Table 3 Data for Control and Treatment Group (3 from 3)

	No.	Group	Elic_Analyst	Analy_Analyst	SpecVal_Analyst
Control Group (Traditional Methodology)	3	0	7.000	6.000	8.000
	4	0	9.000	4.000	9.000
	5	0	5.500	8.500	3.500
	6	0	7.500	5.500	8.000
	8	0	6.000	9.000	19.000
	9	0	6.000	9.000	18.000
	14	0	7.000	7.000	7.000
	17	0	4.000	4.000	5.000
Treatment Group (Novel Methodology)	1	1	6.500	6.500	2.750
	2	1	10.000	3.370	11.000
	7	1	9.500	8.500	1.500
	10	1	10.000	4.000	1.000
	11	1	13.500	7.500	1.000
	12	1	12.000	4.000	2.500
	13	1	15.330	3.080	2.500
	15	1	11.000	5.000	1.500
	16	1	7.750	7.080	1.450

LITERATURE

Adler, R.B., and Rodman, G. *Understanding human communication*, (3rd. ed.) Holt, Rinehart & Winston, London, 1988.

Avison, D., and Fitzgerald, G. *Information Systems Development: Methodologies, Techniques and Tools* McGraw-Hill, 2006.

Baeza-Yates, R.A., and Ribeiro-Neto, B.A. *Modern Information Retrieval* ACM Press / Addison-Wesley, 2002.

Berander, P. "Using Students as Subjects in Requirements Prioritization," Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE'04), 2004, pp. 167-176.

Boehm, B., Grunbacher, P., and Briggs, R.O. "Developing groupware for requirements negotiation: lessons learned," *IEEE Software* (18:3) 2001, pp 46-55.

Boland, R., Tenkasi, R., and Te'eni, D. "Designing Information Technology to Support Distributed Cognition," *Organization Science* (5:3) 1994, pp 456-475.

Browne, G.J., and Rogich, M.B. "An empirical investigation of user requirements elicitation: comparing the effectiveness of prompting techniques," *Journal of Management Information Systems* (17) 2001, pp 223-249.

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

Carmel, E. *Global Software Teams: Collaborating Across Borders and Time Zones* Prentice Hall, Upper Saddle River, N.J., 1999.

Carmel, E., and Tija, P. *Offshoring Information Technology - Sourcing and Outsourcing to a Global Workforce* Cambridge University Press, 2005.

Carver, J., Jaccheri, L., Morasca, S., and Shull, F. "Issues in Using Students in Empirical Studies in Software Engineering Education," Proceedings of the 9th International Software Metrics Symposium (METRICS'03), 2003.

Chen, M., and Nunamaker, J.F. "The Architecture And Design of a Collaborative Environment for Systems Definition," *The DATA BASE for Advances in Information Systems* (22:1-2) 1991, pp 22-28.

Chin, W.W. "The Partial Least Squares Approach for Structural Equation Modeling," in: *Modern Methods for Business Research,* G.A. Marcoulides (ed.), Lawrence Erlbaum Associates, Hillsdale, 1998, pp. 295-336.

Clark, H.H., and Brennan, S. "Grounding in Communication," in: *Perspectives on Socially Shared Cognition,* L.B. Resnick, J.M. Levine and S.D. Teasley (eds.), American Psychological Association, 1991, pp. 127-149.

Cooper, R.B., and Swanson, E.B. "Management information requirements assessment: the state of the art," *The DATA BASE for Advances in Information Systems* (11:2) 1979, pp 5-16.

Coughlan, J., and Macredie, R.D. "Effective Communication in Requirements Elicitation: A Comparison of Methodologies," *Requirements Engineering* (7:2) 2002, pp 47-60.

Cramton, C.D. "The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration," *Organization Science* (12:3) 2001, pp 346-371.

Curtis, B., Krasner, H., and Iscoe, N. "A field study of the software design process for large systems," *Communications of the ACM* (31:11) 1988, pp 1268-1286.

Daft, R.L., and Lengel, R.H. "Organizational Information Requirements, Media Richness and Structural Design," *Management Science* (32:5) 1986, pp 554-571.

Damian, D., Lanubile, F., and Mallardo, T. "On the Need for Mixed Media in Distributed Requirements Negotiations," *IEEE Transactions on Software Engineering* (34:1) 2008, pp 116-132

Damian, D.E., Eberlein, A., Shaw, M.L.G., and Gaines, B.R. "An Exploratory Study of Facilitation in Distributed Requirements Engineering " *Requirements Engineering Journal* (8:1) 2003a, pp 23-41.

Damian, D.E., and Zowghi, D. "Requirements Engineering Challenges in Multi-Site Software Development Organizations," *Requirements Engineering* (8) 2003b, pp 149-160.

Davidson, E.J. "Technology frames and framing: A socio-cognitive investigation of requirements determination," *MIS Quarterly* (26:4) 2002, pp 329-358.

Davis, G.B. "Strategies for information requirements determination," *IBM Systems Journal* (21) 1982, pp 4-30.

Dean, D.L., Lee, J.D., Pendergast, M.O., Hickey, A.M., and Nunamaker, J.F. "Enabling the effective involvement of multiple users: methods and tools for collaborative software engineering," *Journal of Management Information Systems* (14:3) 1997, pp 179-222.

Decker, B., Ras, E., Rech, J., Jaubert, P., and Rieth, M. "Wiki-Based Stakeholder Participation in Requirements Engineering," *IEEE Software* (24) 2007, pp 28-35.

Decker, B., Rech, J., Ras, E., Klein, B., and Höcht, C. "Self-organized Reuse of Software Engineering Knowledge supported by Semantic Wikis," Workshop on Semantic Web Enabled Software Engineering at the 4th International Semantic Web Conference, Galway, Ireland, 2005.

Dibbern, J., Winkler, J., and Heinzl, A. "Explaining Variations in Client Extra Costs between Software Projects Offshored to India," *MIS Quarterly* (32:2) 2008, pp 333-366.

Dutoit, A.H., McCall, R., Mistrik, I., and Paech, B. (eds.) *Rationale Management in Software Engineering*. Springer, 2006.

Ebersbach, A., Glaser, M., and Heigl, R. *Wiki Web Collaboration* Springer, Berlin, 2005.

Edwards, H.K., and Sridhar, V. "Analysis of Software Requirements Engineering Exercises in a Global Virtual Team Setup," *Journal of Global Information Management* (13:2) 2005, pp 21-41.

Egyed, A., and Grünbacher, P. "Identifying Requirements Conflicts and Cooperation: How Quality Attributes and Automated Traceability Can Help," *IEEE Software* (21:6) 2004, pp 50-58.

Fenton, F., Pfleeger, S., and Glass, R. "Science and Substance: A Challenge to Software Engineers," *IEEE Software* (11) 1994, pp 86-95.

Flynn, D.J., and Warhurst, R. "An empirical study of the validation process within requirements determination," *Information Systems Journal* (4:3) 1994, pp 185-212.

Fornell, C. "The Blending of Theoretical Empirical Knowledge in Structural Equations with Unobservables," in: *Theoretical Empiricism: A General Rationale for Scientific Model-Building,* H. Wold (ed.), Paragon House, New York, 1989, pp. 153-174.

Gibson, C.B., and Abstract, J.L.G. "Unpacking the Concept of Virtuality: The Effects of Geographic Dispersion, Electronic Dependence, Dynamic Structure, and National Diversity on Team Innovation," *Administrative Science Quarterly* (51:3) 2006, pp 451-495.

Gotel, O.F., A. "An Analysis of the Requirements Traceability Problem," Proceedings of the 1st International Conference on Requirements Engineering (ICRE'94), IEEE Computer Society Press, 1994, pp. 94-101.

Gruenbacher, P., and Braunsberger, P. "Tool Support for Distributed Requirements Negotiation," in: *Cooperative Methods and Tools for Distributed Software Processes,* A. Cimitile, A.D. Lucia and H. Gall (eds.), Franco Angeli, Milano, Italien, 2003, pp. 56-66.

Gruenbacher, P., and Briggs, R.O. "Surfacing Tacit Knowledge in Requirements Negotiation: Experiences using EasyWinWin," Proceedings of the 34th Hawaii International Conference on System Sciences, 2001.

Grunwald, R., and Kieser, A. "Learning to Reduce Interorganizational Learning: An Analysis of Architectural Product Innovation in Strategic Alliances," *Journal of Product Innovation Management* (24) 2007, pp 369-391.

Guinan, P.J., Cooprider, J.G., and Faraj, S. "Enabling Software Development Team Performance During Requirements Definition: A Behavioral Versus Technical Approach," *Information Systems Research* (9:2) 1998, pp 101-125.

Hannay, J.E., Sjøberg, D.I.K., and Dybå, T. "A Systematic Review of Theory Use in Software Engineering Experiments," *IEEE Transactions on Sodtware Engineering* (33:2) 2007, pp 87-107.

Hayek, F.A. "The Use of Knowledge in Society," *American Economic Review* (35) 1945, pp 519-530.

Heeks, R., Krishna, S., Nicholson, B., and Sahay, S. "Synching' or Sinking: Trajectories and Strategies in Global Software Outsourcing Relationships," *IEEE Software Special Issue* (18:2) 2001, pp 54-60.

Herbsleb, J.D. "Global Software Engineering: The Future of Socio-technical Coordinatioames," Future of Software Engineering, 2007.

Herbsleb, J.D., and Grinter, R.E. "Architectures, Coordination, and Distance: Conway's Law and Beyond," *IEEE Software* (16:5) 1999, pp 63-70.

Herbsleb, J.D., and Mockus, A. "An Empirical Study of Speed and Communication in Globally Distributed Software Development," *IEEE Transactions on Software Engineering* (29:6) 2003, pp 481-494.

Hevner, A.R., March, S.T., Park, J., and Ram, S. "Design Science in Information Systems Research," *MIS Quarterly* (28:1) 2004, pp 75-105.

Hirschheim, R.A., Klein, H.K., and Lyytinen, K. "Exploring the Intellectual Structures of Information Systems Development: A Social Action Theoretic Analysis," *Accounting, Management and Information Technologies* (6:1/2) 1996, pp 1-64.

Hoest, M., Regnell, B., and Wohlin, C. "Using Students as Subjects - A Comparative Study of Students and Professionals in Lead-Time Impact Assessment," *Empirical Software Engineering* (5) 2000, pp 201-214.

Holtzblatt, K., and Beyer, H.R. "Requirements Gathering: The Human Factors - Introduction to the Special Section," *Communication of the ACM* (38:5) 1995, pp 30-32.

IEEE "IEEE recommended practice for software requirements specifications," IEEE STD-830.

Iivari, J., Hirschheim, R., and Klein, H. "A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches," *Journal of Management Information Systems* (17) 2001, pp 179-218.

Iivari, J., and Lyytinen, K. "Research on Information Systems Development in Scandinavia: Unity in Plurality," *Scandinavian Journal of Information Systems* (10:1/2) 1998.

Jimenez-Peris, R., Khuri, S., and Patino-Martinez, M. "Adding Breadth to CS1 and CS2 Courses through Visual and Interactive Programming Projects," Proceedings of the 30th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '99), ACM Press, 1999, pp. 252-256.

Kaiya, H., Horai, H., and Saeki, M. "AGORA: Attributed Goal-Oriented Requirements Analysis Method," Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering (RE'02), IEEE Computer Society Press, 2002, pp. 13-22.

Karlsson, J., and Ryan, K. "A Cost-Value Approach for Prioritizing Requirements," *IEEE Software* (14:5) 1997, pp 67-74.

Katz, A., and Te'eni, D. "The Contingent Impact of Contextualization on Computer-Mediated Collaboration," *Organization Science* (18:2) 2007, pp 261-279.

Keil, M., and Carmel, E. "Customer-developer links in software development," *Communication of the ACM* (38:5) 1995, pp 33-44

Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., Emam, K.E., and Rosenberg, J. "Preliminary Guidelines for Empirical Research in Software Engineering," *IEEE Transactions on Software Engineering* (28) 2002, pp 721-734.

Klein, B., Hoecht, C., and Decker, B. "Beyond Capturing and Maintaining Software Engineering Knowledge - "Wikitology" as Shared Semantics," Workshop on Knowledge Engineering and Software Engineering at Conference of Artificial Intelligence, Koblenz 2005.

Konsynski, B., Kottemann, J., Nunamaker, J., and Scott, J. "PLEXSYS-84: An Integrated Development Environment for Information Systems," *Journal of Management Information Systems* (1:3) 1985, pp 62-104.

Kotonya, G., and Sommerville, I. *Requirements Engineering: Processes and Techniques* Wiley & Sons, 2004.

Krishna, S., Sahay, S., and Walsham, G. "Managing cross-cultural issues in global software outsourcing," *Communications of the ACM* (47:4) 2004, pp 62-66.

Lang, M., and Duggan, J. "A Tool to Support Collaborative Software Requirements Management," *Requirements Engineering* (6) 2001, pp 161-172.

Leonardi, P.M., and Bailey, D.E. "Transformational Technologies and the Creation of New Work Practices: Making Implicit Knowledge Explicit in Task-Based Offshoring," *MIS Quartely* (32:2) 2008, pp 411-436.

Leuf, B., and Cunningham, W. *The Wiki Way: Quick Collaboration on the Web* Addison-Wesley, Boston, USA, 2001.

Lichter, H., Schneider-Hufschmidt, M., and Züllighoven, H. "Prototyping in Industrial Software Projects-Bridging the Gap between Theory and Practice," *IEEE Transactions on Software Engineering* (20) 1994, pp 825-832.

Liou, Y.I., and Chen, M. "Using Group Support Systems and Joint Application Development for Requirements Specification," *Journal of Management Information Systems* (10:3) 1994, pp 25-41.

Lyytinen, K., and Hirschheim, R.A. "Information Systems Failures: A Survey and Classification of the Empirical Literature," *Oxford Surveys in Information Technology* (4) 1987, pp 257-309.

Macaulay, L. *Requirements Engineering* Springer, London, 1996.

Macaulay, L., O'Hare, G., Dongha, P., and Viller, S. "Co-operative requirements capture: prototype evaluation," in: *Computer support for co-operative work,* K. Spurr, P. Layzell, L. Jennison and N. Richards (eds.), John Wiley & Sons, Inc. , New York, NY, USA 1994, pp. 169-194

Majchrzak, A., Beath, C.M., Lim, R.A., and Chin, W.W. "Managing Client Dialogues During Information Systems Design to Facilitate Client Learning," *MIS Quarterly* (29:4), 653-672 2005a.

Majchrzak, A., Birnbaum-More, P., Johnson, J., Sengupta, K., and Wagner, C. "Using Wikis to Generate Learning at ICIS2007," Proceedings of the 29th International Conference on Information Systems, Paris, 2008, pp. 14-17.

Majchrzak, A., Malhotra, A., and John, R. "Perceived Individual Collaboration Know-How Development Through Information Technology-Enabled Contextualization: Evidence from Distributed Teams," *Information Systems Research* (16:1) 2005b, pp 9-27.

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

Mardia, K.V. "Measures of Multivariate Skewness and Kurtosis with Applications," *Biometrika* (57:) 1970, pp 519-530.

Mathiassen, L., Saarinen, T., Tuunanen, T., and Rossi, M. "A Contigency Model for Requirements Development," *Journal of the Association for Information Systems* (8:11) 2007, pp 569-597.

Neill, C.J., and Laplante, P.A. "Requirements Engineering: The State of the Practice," *IEEE Software* (20:6) 2003, pp 40-45.

Nicholson, B., and Sahay, S. "Some Political and Cultural Implications of the Globalisation of Software Development: Case Experience From UK and India," *Information and Organisation* (11:1) 2001, pp 25-43.

Nonaka, I. "Dynamic Theory of Organizational Knowledge Creation," *Organization Science* (5:1) 1994, pp 14-37.

Nosek, J.T., and Schwartz, R.B. "User Validation of Information System Requirements: Some Empirical Result," *IEEE Transactions on Software Engineering* (14:9) 1988, pp 1372-1375

Nunamaker, J., Briggs, R.O., De Vreede, G.-J., and Sprague, R.H.J. "Enhancing Organizations Intellectual Bandwidth: The Quest for Fast and Effective Value Creation," *Journal of Management Information Systems* (17:3) 2000, pp 3-8.

Orlikowski, W.J. "Improvising organizational transformation over time: A situated change perspective," *Information Systems Research* (7:1) 1996, pp 63-92.

Orlikowski, W.J. "Knowing in Practice: Enacting a Collective Capability in Distributed Organizing," *Organization Science* (13:3) 2002, pp 249-273.

Peffers, K., Tuunanen, T., Rothenberger, A., M., and Chatterjee, S. "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems* (24:3) 2008, pp 45-77.

Robbins, J. "Adopting Open Source Software Engineering (OSSE) Practices by Adopting OSSE Tools," in: *Free/Open Source Processes and Tools,* J. Feller, B. Fitzgerald, S.A. Hissam and K.R. Lakhani (eds.), MIT Press, 2005, pp. 245-264.

Robertson, J., and Robertson, S. *Requirements-Led Project Management: Discovering David`s Slingshot* Addison Wesley Boston, USA, 2005.

Robertson, J., and Robertson, S. *Mastering the Requirements Process*, (2nd ed.) Addison Wesley, Upper Saddle River, USA, 2006.

Robertson, S. "Requirements for Managing Requirements: Agile Product & Project Management Advisory Service," *Cutter Consortium, Executive Report* (8:9) 2007, pp 1-21.

Robinson, W.N., Pawlowski, S.D., and Volkov, V. "Requirements Interaction Management," *ACM Computing Surveys* (35:2) 2003, pp 132-190.

Rodriguez, F., Geisser, M., Berkling, K., and Hildenbrand, T. "Evaluating Collaboration Platforms for Offshore Software Development Scenarios," Proceedings of the 1st International Conference on Software Engineering Approaches For Offshore and Outsourced Development, Springer, 2007, pp. 96-108.

Ruhe, G., Eberlein, A., and Pfahl, D. "Quantitative Win-Win - A New Method for Decision Support in Requirements Negotiation," Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, ACM Press, 2002, pp. 159-166.

Ruhe, G., Eberlein, A., and Pfahl, D. "Trade-Off Analysis for Requirements Selection," *International Journal of Software Engineering and Knowledge Engineering* (13:4) 2003, pp 345-366.

Saaty, T.L. *The Analytic Hierarchy Process* McGraw Hill, New York, 1980.

SEI "CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing," Version 1.1, Continuous Representation (CMMI-SE/SW/IPPD/SS, V1.1, Continuous), Carnegie Mellon Software Engineering Institute.

Seyff, N., Kroiher, E., Hoyer, C., and Grünbacher, P. "Enhancing GSS-based Requirements Negotiation with Distributed and Mobile Tools," Proceedings WETICE'05: 14th IEEE

International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005, pp. 87-92.
Sinha, V., Sengupta, B., and Chandra, S. "Enabling Collaboration in Distributed Requirements Management," *IEEE Software* (23:5) 2006, pp 52-61.
Sjoberg, D.I., Hannay, J.E., Hansen, O., Kampenes, V.B., Karahasanovic, A., Liborg, N., and Rekdal, A.C. "A Survey of Controlled Experiments in Software Engineering," *IEEE Transactions on Software Engineering* (31) 2005, pp 733-753.
Smith, S., Lai, L., and Khedri, R. "Requirements Analysis for Engineering Computation: A Systematic Approach for Improving Reliability " *Reliable Computing* (13:1) 2006, pp 83-107.
Sommerville, I. *Software Engineering 7* Addison-Wesley, 2004.
Sommerville, I., Sawyer, P., and Viller, S. "Viewpoints for Requirements Elicitation: A Practical Approach," Proceedings of the 3rd IEEE International Conference on Requirements Engineering (ICRE'98), IEEE Computer Society Press, 1998, pp. 74-81.
Spence, R. *Information Visualization: Design for Interaction* Prentice-Hall, 2007.
Sutcliffe, A. "Scenario-Based Requirements Analysis," *Requirements Engineering* (3:1) 1998, pp 48-65.
Te'eni, D. "Review: A cognitive-affective model of organizational communication for designing IT," *MIS Quartely* (25:2) 2001, pp 251-312.
Teasley, S.D. "Rapid Software Development through Team Collocation," *IEEE Transactions on Software Engineering* (28:7) 2002, pp 671-683.
Teichrow, D., and Hershey, E.A. "PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems," *IEEE Trans. on Software Engineering* (3:1) 1977, pp 41-48.
Toval, A., Nicolas, J., Moros, B., and Garcıa, F. "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach," *Requirements Engineering* (6:4) 2002, pp 205-219.
Trevino, L.K., Lengel, R.H., and Daft, R.L. "Media Symbolism, Media Richness, and Media Choice in Organizations: A Symbolic Interactionist Perspective," *Communication Research* (14:5) 1987, pp 553-574.
Valusek, J.R., and Fryback, D.G. "Information Requirements Determination: Obstacles Within, Among, and Between Participants," in: *Information Analysis: Selected Readings,* R. Galliers (ed.), Addison Wesley, Reading, MA, 1987, pp. 139-151.
van Lamsweerde, A. "Requirements Engineering in the Year 00: A Research Perspective," Proceedings of the 22nd International Conference on Software Engineering, ACM Press, Limerick, 2000.
van Lamsweerde, A., Letier, E., and Darimont, R. "Managing Conflicts in Goal-Driven Requirements Engineering," *IEEE Transactions on Software Engineering* (24:11) 1998, pp 908-926.
Vitalari, N.P. "Knowledge as a Basis for Expertise in Systems Analysis: An Empirical Study," *MIS Quarterly* (9:3) 1985, pp 221-241.
Walls, J.G., Widmeyer, G.R., and Sawy, O.A.E. "Building an Infomation System Design Theory for Vigilant EIS," *Information Systems Research* (3:1) 1992, pp 36-59.
Walz, D.B., Elam, J.J., and Curtis, B. "Inside a software design team: knowledge acquisition, sharing, and integration," *Communications of the ACM* (36:10) 1993, pp 63-77.
Westrup, C. "Knowledge, legitimacy and progress? Requirements as inscriptions in information systems development," *Information Systems Journal* (9 1) 1999, pp 35-54.
Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., and ., A.W. *Experimentation in Software Engineering: An Introduction* Kluwer Academics Publications, 2000.
Zigurs, I., and Buckland, B.K. "A Theory of Task/Technology Fit and Group Support Systems Effectiveness," *MIS Quarterly* (22:3) 1998, pp 313-334.