IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group

ITP4850

Supporting Knowledge Exchange Isn't Easy: Lessons Learnt From A Case Study

Dulce T. Pumareja[1], Klaas Sikkel[1] and Tatyana Bondarouk[2]
[1]Faculty of Electrical Engineering and [2]Faculty of Business, Public Administration & Technology
University of Twente
PO Box 217, 7500 AE Enschede, the Netherlands
tel.: +31-53-4894283, tel.: +31-53-4893666
fax: +31-53-4892927, fax: +31-53-4892159
{pumareja|sikkel}@cs.utwente.nl, *t.bondarouk@sms.utwente.nl*

ABSTRACT

A knowledge management system is introduced in a large insurance company. It is meant to become a virtual knowledge network for a group of insurance professionals. Despite the fact that the introduction was met with enthusiasm and user participation in the design was ensured, the system did not live up to the expectations. In this paper we investigate this failure by uncovering and analyzing the requirements engineering processes underlying the system's conception, design and introduction. The demise of the system can be attributed to a lack of attention for the use context of the end users and a tendency in the development process to focus mostly on the technical solution instead of problem analysis.

INTRODUCTION

The reality remains that many IT implementation projects are motivated by a technology-push initiative in order to achieve business and organizational goals. Usually, IT implementation proceeds with already a pre-conceived IT solution in mind, without taking into thoughtful consideration the problem context and the needs of the people who would use the system. Involving future system users in IT implementation projects can help mitigate certain risks and is acknowledged to be a key factor for a successful implementation (Lin & Shao, 2000; Hunton & Beeler, 1997). However, in practice it does not guarantee that a system will be used as expected. Users can fully agree and accept the *idea* of a system, but working with it will still be problematic.

In this paper we describe a case study from the field of knowledge management (KM). Information technology plays an important role in knowledge management. The availability of technologies such as Lotus Notes and WWW has been instrumental in catalyzing the knowledge management efforts of organizations (Kock, 2000; Ciborra, 1996). But KM cannot be brought about by technology alone. It has to fit the organizational context and culture, otherwise it won't work (Davenport & Prusak, 1998).

The case study aimed at understanding the following dilemma. The users were involved in the implementing a system designed to support knowledge sharing. They participated in decision-making, functional design and system piloting. They were fully informed about the project, delivered the authority to give advice and make decisions. However, system use fell very much short of the expectations.

We sought to explain the failure of the system from the perspective of requirements engineering. We believe this perspective allows us to deepen our understanding of how such failures occur by analyzing how system functions and constraints relate to real world needs and goals.

REQUIREMENTS ENGINEERING

Requirements Engineering (RE) is the part of software engineering concerned with exploring the goals, functions, and quality properties of a system – collectively known as requirements, and maintaining these throughout the system's lifecycle. More precisely: RE is a purposeful set of activities aimed at discovering, learning, understanding and maintaining a set of requirements for a computer-based system (Kotonya & Sommerville, 1998).

The lack of attention to or a superficial treatment of this process has been recognized as the major reason as to why IT systems fail in meeting expectations and goals regardless of the application domain (Lauesen, 2001; Gause & Weinberg, 1989). These failures can be translated in terms of missing functionality, ill-defined system mission, poor interface and inefficient user and task support. This holds for KM as much as any IT implementation domain. Consequently, these issues comprise the domain of RE and should have been resolved by an optimal RE process.

RE is basically the problem analysis phase of a software design process (Wieringa, 2003, 1996). It consists of both static and dynamic elements, which we would refer to as constructs and processes accordingly.

RE Constructs

These are generic elements in any practical problem solving, which include (Wieringa, 2003, 1996):

- *Phenomena*: the problem setting, situation and conditions surrounding the problem context.
- *Stakeholders*: the people and organizations whose needs have to be satisfied.
- *Goals*: the requirements to be satisfied.
- *Inhibitors*: the constraints that limit design freedom.

RE Processes

As an engineering effort, problem analysis and solution specification in RE are systematically tackled through a generic set of activities (Nusebeih & Easterbrook, 2000):

- *Requirements elicitation*: data gathering aimed at learning, discovering and surfacing the needs of stakeholders.
- *Modeling and analyzing requirements* in order to increase understanding and to check for requirements conflict, inconsistency and incompleteness.
- *Communicating requirements* to stakeholders in order to secure a common understanding.
- *Agreeing requirements*: the process of negotiation to handle conflicting requirements, set priorities and manage stakeholder expectations.
- *Specification*: the formalization and documentation of the chosen software design solution for purposes of traceability and management.

RE is a highly iterative and evolutionary process in which the activities are not bounded to a sequential one-shot effort (Pfleeger, 1999). There is no one best way of doing RE nor is there one ideal RE process that fits all instances of problem solving. What remains invariant however is the idea that in order to build or acquire systems that the intended people will use, and like to use, it has to be founded on a profound understanding of needs, tasks, goals and operating contexts. To that effect, the generic RE framework above can be applied.

CASE STUDY AND METHODS

The study was conducted at *InsurOrg*, one of the largest insurance companies in the Netherlands with 12,500 employees all over the country. Our paper describes an IT implementation project in the Non-Life Insurance division of *InsurOrg*, namely, the introduction of a digital Knowledge Network – *KnowledgeNet*.

The study was conducted for a period of six months. This was done seven months after system installation when it had become evident that the system did not live up to expectations.

Data collection was done mostly through qualitative methods: interviews, observations, document analysis and system inspection. A total of 25 interviews were conducted. The following documents were analyzed: Annual Report 2001, system manual, *KnowledgeNet* business plan, project plan and system log archives.

To address the question of how the failure came about, we have reconstructed the requirements analysis and design process of *KnowledgeNet* using the RE framework outlined in the previous section. The underlying hypothesis is that an optimal RE process should have led to a successful system; hence it should be possible to relate obstacles for successful use to specific RE errors along the way. Although it is an *ex-post* reconstruction, the interviews and the documents yielded enough information to uncover the relevant aspects of both RE constructs and processes.

SYSTEM DESCRIPTION

KnowledgeNet is a distributed information system for supporting knowledge exchange. It is a repository of general-purpose information locally referred to as "knowledge items". It is a custom-made system developed using Lotus Notes. Its key functional properties, in terms of what it allows users to do, include specific information search, search for domain experts, publish, edit, comment and acquire information from a database. Its application structure can be described in terms of its two main information repositories: a knowledge database and a domain-experts directory. These two are made accessible by a navigation portal, from which other parts and services of *KnowledgeNet* can also be accessed. Further, users must have Lotus Notes accounts and clients installed onto their computers in order to use it.

RECONSTRUCTING *KNOWLEDGENET*

RE Constructs

Phenomenon

The company, *InsurOrg*, is a result of mergers and acquisitions. Its up-to-date organizational structure is rather complex. All formerly independent sub-companies still operate under their original business labels, but there were efforts aimed at reinforcing the merging processes. The fusion faces complex organizational changes due to the differences in how management and business is carried out in each sub-company. In order to benefit from the expertise of each sub-company, *InsurOrg* took on the strategy of cooperation, bringing all employees together and creating a new cooperative organizational structure. Such was the driving force to implement Knowledge Management as a strategy to achieve new organizational development.

In the Non-Life Insurance Division, KM strategies are realized through the competence center, the Knowledge Center for Non-Life Insurance (KCNLI) whose responsibilities are to:

- Develop and maintain non-life insurance knowledge;
- Lead projects for the division;
- Stimulate knowledge creation and sharing;
- Build a community of non-life insurance experts.

To develop common competence, KCNLI started a long-term Knowledge Network project aimed at structuring, initiating and organizing knowledge sharing within the non-life insurance group. This project consisted of two parts:

- regular face-to-face meetings/workshops
- a virtual knowledge network IT system

The knowledge management system *KnowledgeNet* is a realization of the second part of the project.

Stakeholders

There are three stakeholder groups assuming three different roles in the IT implementation:

- Knowledge Center KCNLI as Project Champion
- Non-Life Insurance Experts as Targeted End-Users
- IT Department as System Designer and Developer

KCNLI consists of five persons including a manager who carries the final responsibilities for the center's functions. One of the KCNLI staff is the project leader of *KnowledgeNet*.

The non-life insurance experts are the intended end-users forming a total of 33 employees representing five different sub-companies in five different locations. The experts consist of two types of professionals: product managers (22) and actuaries (11).

Product managers are responsible for the management and development of non-life insurance products. These products are classified in three groups: mobility (private cars, motorbikes, caravans, trucks, lories, etc.), recreation (boats, yachts, travel, etc.), home insurance (valuables, legal services, glass, fire damage, third party liability, etc.). Their primary tasks include: market analysis, legislation procedures, new products development, and knowledge 'monitoring' across the whole company regarding non-life insurance products.

Actuaries (insurance mathematicians) are responsible for the statistical analysis of company benefits from its non-life insurance products. Their tasks include: risk analysis, calculations and analysis of premiums, reserves calculations and analysis, and re-insurance strategy analysis.

One software engineer who developed *KnowledgeNet* represented the IT Department.

Goals

System goals as identified by KCNLI were:

- *Short-term goal*. To provide technical support for creating, gathering and disseminating knowledge. *KnowledgeNet* is to become the 'spot' where information could be recorded, collected, structured, ordered, stored, retrieved and exchanged.
- *Long-term goal*. To develop common knowledge as the opposite to shared information. The system is to contribute to the unified expertise of the specialists.
- *Ultimate goal*. To support community building through the development of common knowledge. Becoming one team across five sub-companies is the most important issue for KCNLI.

Inhibitors

Lack of budget is the biggest obstacle in the process, as *KnowledgeNet* is a local initiative from KCNLI. This precluded any attempt at acquiring any additional applications and constrained design decisions: Lotus Notes, already an existing infrastructure at *InsurOrg* used for email and shared databases, is to be the development environment for KnowledgeNet.

RE Processes

The design and development of KennisNet was performed in two iterations. Two versions of the system were developed, one in 1998 and the most recent in 2001. The initial system was an empty database shell

that allowed users to input information. It was put into use for a limited number of users. Eventually KCNLI decided to make major improvements to the system, for a more widespread use, prompting a second round of requirements iteration.

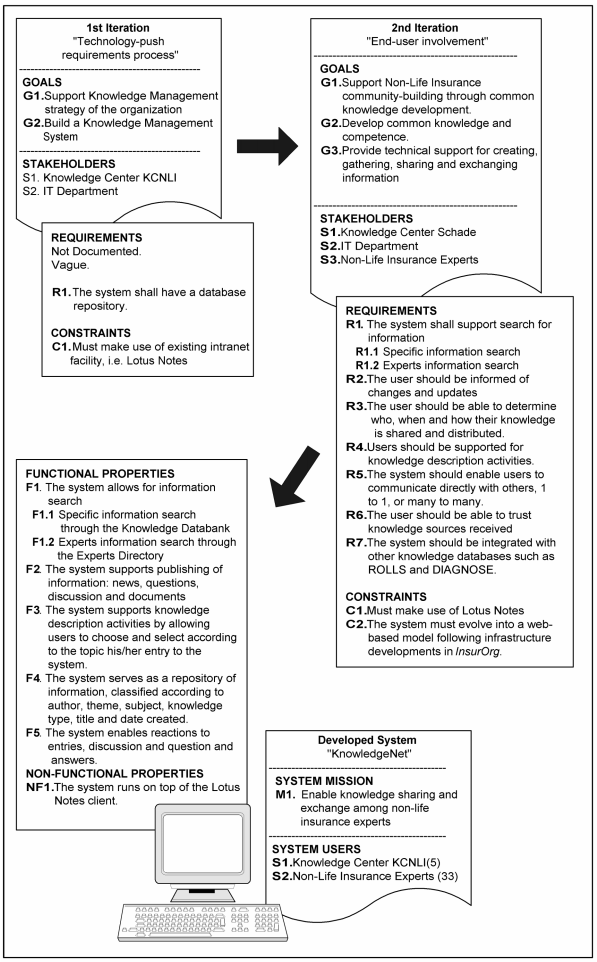
Requirements Elicitation

Requirements elicitation took place in two iterations. In the first iteration, requirements were identified through a series of brainstorming sessions within KCNLI. They discussed and communicated with the IT Department about what kind of system they would like to have.

In the second iteration, KCNLI decided to include end-users in designing *KnowledgeNet*. To do so, KCNLI introduced the idea of developing a knowledge management system to the experts in one of the workshops. They were asked whether such a system would be desirable for them. A positive reply marked the beginning of a participatory design process.

Several experts were asked to represent the division in a series of workshops aimed at deriving the functional properties that *KnowledgeNet* must have. This participatory process led to the formulation of a more precise set of requirements for the new *KnowledgeNet*. Figure 1 elaborates on this process of evolution, emanating from a vague set of requirements, governed mainly by the desire to create a system without a clear idea of what it is supposed to do, to a more definite set of requirements coming from the targeted system users.

Figure 1. The KnowledgeNet Requirements Process at InsurOrg



Modeling and Analyzing Requirements

No formal models of the requirements were formulated from the requirements elicitation workshops. However, one concrete output of the design workshops was the creation of a classification scheme for organizing the contents of the knowledge database. The classification scheme served as a cognitive guide for describing knowledge in terms of topics that were meaningful for the experts as these were derived from their own domain.

Communicating Requirements

Most of the communications took place in the second iteration. The design workshops and meetings served as venues for communication. However, communication was mainly oriented towards promoting the idea of a KM system, and whether it was a desirable solution and what kind of functionalities the system should have.

Agreeing Requirements

An enthusiastic reply from future end-users about the idea of implementing a knowledge exchange system prompted KCNLI to proceed with the re-design and development of the first *KnowledgeNet*, without seeing the need for further negotiations. There were no further checks done on the requirements to see which were feasible or not. As a result, some requirements were not implemented due to budget and technical limitations. But these were not communicated or explained further to the users.

Specifying Requirements

In both instances, no software requirements documents were produced to guide the development of *KnowledgeNet*. At best, the documentation available about the project is the business and project plan documents prepared by KCNLI.

SYSTEM USAGE

Highlights of *KnowledgeNet* use derived and analyzed from system logs reveal the following:

- Less than half (46%) of the user base (N=39) have made an entry to the system;
- 67 new entries were added to the system, an average of 9.6 entries per month;
- 30 entries are official insurance reports of 20-40 pages; seven entries are 'questions' and even less – four, are 'discussion' items;
- 31% of all the logged contributions to the database (N=143) are contributed by only one person, who happens to be the project leader for the *KnowledgeNet* project;
- Lotus Notes system activity logs recorded 800 user activities related to reading and searching for information within *KnowledgeNet*.

These figures indicate that users preferred only one side of knowledge exchange: they used *KnowledgeNet* mainly to read information. End-users perceived the system with a search engine metaphor in mind. They view the system as an information repository rather than as a knowledge sharing space. System interaction is motivated by information search impulses, more than by the awareness to share information with colleagues. This mental model of system use is the reason for non-use and breakdown situations. Users cited the limited contents of the database as the reason why they do not use or have stopped using the system.

On the other hand, making a contribution to the system is not an easy straightforward process – there are certain organizational obstacles. In some sub-companies, publishing reports still require clearance from the experts' managers before it can be made available to other sub-companies within the same division.

KnowledgeNet use is voluntary. However, making a contribution is an extra task. Users were encouraged and forced to publish documents. The project leader made relentless requests about making contributions. These were made visible in the start-up page of *KnowledgeNet* itself. Requests for contribution were further communicated through emails,

and formed part of the agenda in the regular business meetings between KCNLI and non-life insurance experts.

DISCUSSION

From the RE reconstruction account, the following are points for analysis and discussion:

RE Analysis: Rather Limited

The development of *KnowledgeNet* proceeded on the basis of an ad-hoc requirements practice. The initial requirements for *KnowledgeNet* were derived from a vague goal not related to the needs of end-users. The initial requirements came only from one stakeholder group, KCNLI. During the second iteration, more requirements were identified and specified based upon the users' point of view. However, the *elicitation* was still focused on a techno-centric point of view. Attention was not given to understanding detailed task identities of the users, their work cultures, traditions and responsibilities. Other steps of the RE process were inadequate to the desired situation. We found the main limitation as absence of professional discussions with the users: the use of the system in all its aspects was hardly communicated. In other words, *analyzing, communicating, and agreeing* steps became the weakest issue in the design process. Finally, we did not find any specification document to trace the process.

System goals are not transferred to real world needs

Technically, the system fulfills some of its goals as it supports information exchange. But the underlying assumption, that users are willing to share knowledge via the system, was not studied carefully.

System developed does not meet essential user requirements

There are some user requirements the current system does not meet. Among these include the need for integration with other local databases and with the WWW. Also crucial for the users is a notification function for new items in the system. The current *KnowledgeNet* does not yet fulfill these requirements.

CONCLUSIONS

In this case study we have analyzed the experiences of an organization implementing a system for knowledge sharing. System design draws upon knowledge from the areas of knowledge management and user-centered design. The users met the idea of introducing the system with enthusiasm, because they felt a need for supporting their exchange of knowledge and expertise. However, after the system was installed, it was hardly used.

There are two general issues that stand out and are worth repeating here, because they are general risks to such projects.

Firstly, for a successful implementation the system was critically dependent on the cooperation within the end-user group. Hence the design process should have focused on their needs and desires. The system was conceived to accommodate the needs of different stakeholders, while in practice it mainly served KCNLI.

This may have been the cause of the second major failure: the context in which the system was to be used has not been properly addressed. The requirements analysis concentrated on the technical properties of the system, the desired functionality. It was overlooked that the affordances of the system, even if the user interface had been perfect, did not fit well into the everyday tasks of the users.

If a system is designed in an exploratory context with limited resources, it is usual that not all requirements are specified at an earlier stage. In such a case it is important to make explicit choices about prior requirements for the current stage of RE. Otherwise the project might result in an ad-hoc development process, where only the 'easiest' requirements, realizable within the available technical resources are implemented.

A comprehensive requirements process should include both the technical system and the work situation in which the system is to be deployed. A good understanding of how the users do their tasks is necessary, especially if they work as a group. In the area of knowledge management, this includes how they communicate, search and acquire information from colleagues and other sources.

REFERENCES

Ciborra, C. (1996). *Groupware and Teamwork: Invisible Aid or Technical Hindrance*. John Wiley, Chichester.

Davenport, T.H. & Prusak, L. (1998). *Working Knowledge – How Organizations Manage What They Know.* Harvard Business School Press.

Gause, D. & Weinberg, G. (1989). *Exploring Requirements: Quality Before Design*. Dorset House Publishing, New York.

Hunton, J. & Beeler, J. (1997). Effects of User Participation in Systems Development: A Longitudinal Field Experiment. *MIS Quarterly*, December, 359-383.

Kock, N. (2000). Sharing Interdepartmental Knowledge Using Collaboration Technologies. *Journal of Information Technology Impact*, 2(1), 5-10.

Kotonya, G. & Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*. John Wiley, Chichester.

Lauesen, S. (2002). *Software Requirements: Styles and Techniques*. Addison-Wesley.

Lin, W. & Shao, B. (2000). The Relationship Between User Participation and System Success: A Simultaneous Contingency Approach. *Information & Management*, 37, 283 - 295.

Nuseibeh, B. & Easterbrook, S. (2000). Requirements Engineering: A Road Map. *Proceedings of International Conference on Software Engineering (ICSE-2000)* Limerick, Ireland: ACM Press.

Pfleeger, S.L. (1999). *Software Engineering: Theory and practice*. Prentice-Hall International, Singapore.

Wieringa, R. (2003). *Design Methods for Reactive Systems: Yourdon, Statement and the UML*. Morgan Kaufman, Los Altos, CA.

Wieringa, R. (1996). *Requirements Engineering: Frameworks for Understanding*. Wiley, Chichester.