Groupware Support for Requirements Management in New Product Development

Ahti Salo
Systems Analysis Laboratory
Helsinki University of Technology
02150 HUT, Finland
Ahti.Salo@hut.fi

Timo K. Käkölä
Department of Computer Science and Information Systems
University of Jyväskylä
40351 Jyväskylä, Finland
timokk@cc.jyu.fi

June 2002

Groupware Support for Requirements Management in New Product Development

Abstract

Large high-technology companies operate in fiercely competitive international markets. To succeed they need to shorten the cycle time of new product development (NPD) while improving product quality and maintaining or reducing the total resources required. Their abilities to meet these business goals depend on how extensively and effectively they collect, analyze, and utilize requirements in their product development. Creating and sharing such knowledge is complicated partly because the NPD activities of large companies are geographically distributed. Groupware technologies allow knowledge to be created and shared more effectively. Thus they hold considerable potential as means of meeting the goals. Yet, little theory-based guidance is available to help design groupware-based requirements management systems (RMS) for large geographically distributed organizations. This paper draws on existing literature and experiences from large-scale distributed industrial development projects at Nokia to start building a design theory that answers the following question: What are the necessary and sufficient properties of RMS and how RMS should be designed and introduced in large high-technology companies to best achieve the business goals? The main contributions of the paper are (1) the creation of a generic set of requirements for RMS, (2) the development of the RMS design, and (3) the empirical validation of the design by analyzing instantiations of the design at various product lines of Nokia.

Keywords

Computer-mediated communication; Decision support systems; Groupware; Information systems design theory; Knowledge management; New product development; Organizational learning; Process maturity; Requirements management; Shared knowledge creation.

I. Introduction

The development of high technology products is characterized by pressures towards shorter time-to-markets, increasing complexity of product designs, globalization of markets, and continuous price erosion. To succeed under these conditions, high-tech companies need to shorten the cycle time of new product development (NPD) while improving product quality and maintaining or reducing the total resources required. Their abilities to meet these business goals depend on how extensively and effectively they collect, analyze, and utilize requirements in their product development. It is crucial that high-tech companies help their personnel become more knowledgeable about the marketplace and leverage the accumulated knowledge effectively [15,26,30-31,37,43]. This is particularly true during the earliest phases of NPD in which different functions – marketing and R&D in particular - need to integrate their knowledge into a product concept that provides direction for the downstream phases of NPD [8]. Indeed, there is plenty of evidence that knowledge integration is a key determinant of successful product development [7,10].

Yet, the achievement of such integration is complicated by several factors. In large companies the development activities are often scattered across multiple sites, which limits possibilities for setting up face-to-face meetings. Moreover, differences in organizational culture and divergent perceptions about the product's mission may make it difficult to reach an agreement about how the prospective product should be defined [8]. Against this background, communication, coordination, and collaboration support for the early phases of NPD activities poses a significant challenge with substantial payoffs.

The rapid progress of networking technologies has offered new possibilities for the creation and sharing of knowledge within organizations. Groupware systems for co-ordination and collaboration, in particular, have helped organizations increase the productivity of their knowledge-intensive processes. It is the quest for these productivity gains, which has spurred multinationals to invest in networked information technology (IT) infrastructures on top of which knowledge-reliant processes may be run.

Few, if any, studies have investigated holistically (1) which processes within NPD are amenable to networked IT solutions, (2) what targets should be set for such solutions, and (3) how these targets can be translated into (A) prescriptive, widely applicable, and theoretically grounded design processes and information system designs and (B) concrete implementations that not only meet the targets but enable the refinement and validation of the design processes and conceptual designs as well as their underlying theories. For example, Hameri and Nihtilä [19] consider the intranet-enabled use of product specifications in a large NPD project and note that prior research on networked NPD-processes is very limited. Gorton, Hawryszkiewycz and Fung [17] report positive results from groupware support in their experimental research, but are uncertain as to whether their findings from a small scale environment apply to large development projects. Herlea and Greenberg [21] present a groupware application for requirements engineering (RE) but note that their research on the use of the room metaphor in RE is only a start towards a better understanding of how groupware can support distributed RE processes. Thus, there is a need for further research on how networked IT solutions can be designed to enhance the effectiveness of industrial NPD processes in multi-site and multi-organizational settings.

The focus of this paper lies at the intersection of computer-supported collaborative work and RE. Specifically, the emphasis is on how groupware-based requirements management systems (RMS) can be designed and used to redesign the earliest phases of product development in multi-site, cross-functional NPD organizations. This RMS-enabled process development should help high-tech companies meet the business goals and respond to the challenges identified above.

Design theories, unlike other theories, support the achievement of goals [52]. Walls, Widmeyer, and El Sawy [52, p. 37] argue that the information systems (IS) "field has now matured to the point where there is a need for theory development based on paradigms endogenous to the area itself" and call for information system design theories (ISDT) to fulfil that need. An ISDT is "a prescriptive theory based on theoretical underpinnings which says how a design process can be carried out in a way which is both effective and feasible" [52, p. 37].

This paper serves as the first stage in building an ISDT for RMS in NPD organizations. It builds on a three-stage research project that was conducted from 1995 to 1999 at Nokia. In the first stage, we helped Nokia's product lines redesign their requirements management (RM) processes and migrate them onto a groupware platform. In designing these groupware-supported RM processes, we collaborated with key personnel from R&D and product marketing who took an active part in the status-quo analysis, target setting and, later on, the institutionalization of RM processes. In the second stage – which was carried out in parallel with this constructive research work – we conducted a literature review on the use of groupware in NPD. The results of this review helped in the design of a groupware-supported RM process and lead us to believe that our work is of relevance to other NPD settings as well because many of findings within Nokia were aligned with earlier accounts in the research literature. Key deliverables from these two stages were (1) the creation of a generic set of requirements for RMS, (2) the development of the conceptual RMS design meeting the requirements, and (3) the domain specific instantiations of the design in different product lines of Nokia. Hereafter, we use the terms "RMS" and "RMS artefact" to refer, respectively, to the RMS design and an instantiation of RMS.

Finally, we also conducted an in-depth follow-up study at five business units of Nokia to validate and refine RMS and to assess empirically the implications of RMS on the RM processes. Towards this end, a total of 30 interviews were conducted with product and marketing managers, R&D engineers and managers, and groupware and information management specialists [47]. This third stage was conducted over a one-year period when RMS had been introduced in about 15 product lines at Nokia. In this paper, we report some lessons from this validation of RMS, although the main focus is on the first two stages of the research project.

The paper is structured as follows. Section II addresses the potential of groupware in NPD, describes the research context at Nokia, characterizes design theories, and presents definitions of the components of an ISDT. Section III discusses requirements on RM tools in distributed NPD organizations. Section IV translates these requirements into the conceptual design of RMS. Section V considers the deployment of RMS and discusses the implications of RMS on NPD to assess the validity of the RMS design. Conclusions and research topics for future development and validation of the ISDT are given in the last section.

II. Groupware Support for Requirements Management

While the growth of groupware has been staggering in the 1990's, much of groupware application development has focused on administrative and operational processes - such as order processing and purchasing – that are relatively well structured [27]. However, advances in commercially available groupware platforms (e.g., improved robustness, usability, and functionality) have made these platforms increasingly viable options also for the redesign and implementation of NPD processes which, by their very nature, are knowledge-intensive and ill-structured [14,27].

The need for cross-functional communication is particularly critical in requirements management which covers (1) the systematic collection of information about customer needs, technical constraints and, more generally, any information which needs to be accounted for in product decisions, (2) the refinement of such information into representations that are suitable for systematic assessments within the NPD organization, and (3) the preparation and recording of product decisions as part of the earliest milestone reviews in product development [46].

Much of the earlier literature suggests that RM processes, perhaps more than other NPD processes (e.g., specification, system testing), are amenable to groupware support. First, the need to establish good communication and collaboration between different functional groups is often highest in requirements capture and analysis [25]. Second, this phase - sometimes referred to as the "fuzzy front end" - belongs to the largest and cheapest opportunities to shorten the development cycle [39]. Third, groupware contributes to effective knowledge sharing and integration, which are both processes that advanced software development environments should support [12]. Fourth, many of the problems during the later phases of the product life-cycle – such as the need for rework and failures in system integration – can be attributed to fluctuating and conflicting requirements due to communication and co-ordination breakdowns [12]. Against this background, groupware appears a promising platform for RM processes in NPD organizations with multi-site and cross-functional development activities.

By supporting the management of codified knowledge[1], groupware also contributes to the creation of an organizational memory, that is, "the means by which knowledge from the past is brought to bear on present activities, thus resulting in higher or lower levels of organizational effectiveness" ([40, p. 89]; see also [31-32]). This is crucial because products such as telecommunications equipment are often delivered as incremental releases to earlier versions of the products, wherefore the NPD organization needs to retrieve up-to-date information about the features that have been supplied in earlier releases or, alternatively, are planned for future releases. At best, tools for RM support can also help the NPD organization raise the maturity of its project planning practices, for instance by encouraging the collection of information about estimated and actual data from earlier projects (e.g., milestones, work effort). Moreover, since groupware tools allow information to be retrieved with few restrictions on time and place, they reduce organizational dependence on immediate access (e.g., phone, face-to-face meetings) to the expertise of individual experts.

A. The Research Setting at Nokia

At Nokia, the motivation for developing a groupware-based RMS to support the earliest phases of NPD arose from (1) the perceived need – as expressed by middle-level managers in product marketing and R&D – to establish effective communication support for requirements processes and from (2) the availability of a global groupware infrastructure which was seen as a suitable platform for such communication. This is not to say that other processes in NPD would not have been suitable for groupware support. Rather, it

[1] Tacit knowledge rooted in human intentional action, while an important component of organiza-

was felt that the payoff from such support would be highest in the context of RM processes, given the geographical dispersion and extensive communication needs of the stakeholders.

The main objective of the RMS development was to implement robust, scaleable, and effective processes for the activities, which precede the earliest milestone reviews. These activities include (1) the capture of new product ideas from product marketing, R&D, and other relevant sources as an input to the RM process; (2) the conversion of such ideas into well-defined expressions of product functionality; (3) the assessment of proposed functionalities in view of their market potential, technical feasibility, compliance with product strategy, and other factors that are relevant to product decisions; (4) the recording of go/no-go decisions about the product functionalities; (5) systematic monitoring of the process by all stakeholders; and (6) the preservation and dissemination of the data resulting from the exchange of ideas on the electronic media.

The development of RMS was organized as a joint project between the Nokia Research Center (NRC), the corporate research unit of Nokia, and Nokia's business units (BU)[2]. Initially, two BUs from Nokia Telecommunications and one from Nokia Mobile Phones were involved in RMS development, but after the first eight months the number of BUs grew to five. Each BU appointed a project representative (typically a person with very good knowledge about the NPD process, e.g., quality manager) who then collaborated with NRC on the assessment of RM needs, the specification of enhanced processes, the design of RMS and, later on, the institutionalization of groupware-supported RM processes.

The BUs faced considerable challenges in developing telecommunications products which were (1) aimed at a well-defined marketplace or market segment and (2) defined in terms of features so that – from the customer's point of view – each feature was an implementation of some desired functionality. A key objective of the groupware-supported RM process, therefore, was to support the generation and selection of new product features subject to the constraints imposed by product strategy, available resources, and compatibility restrictions (e.g., choice of platform; see [29]).

At the beginning of RMS development, an extensive round of interviews was conducted with some 40 representatives from the different BUs. The existing documentation structure was also analyzed in order to better understand RM processes as they were. These analyses produced ideas, which supported the development of early RMS pilot implementations. These implementations were first taken into limited operational use at four BUs

---

tional memory, is beyond our scope [30-31].

[2] At NRC, the first author was the manager of the project at the time when the RMS was developed.

for a period of some six months during which suggestions for improvement were made and integrated into RMS (see section IV). Once the BUs were convinced of the benefits of groupware support, RMS was effectively frozen and the RMS artefacts were deployed into large-scale operational use. By this time, RMS development had taken about one and a half years, whereby the adoption of the artefacts proceeded in parallel with the build-up of a corporate-wide Lotus Notes™ infrastructure.

The participation of several BUs was instrumental in designing a generic, productized RMS solution which was eventually deployed to more than a dozen product lines by the end of the 1990s. First, the involvement of several BUs produced more ideas than what would have been gained from discussions with a single BU. Second, the collaboration with many BUs through the development of pilot RMS implementations helped differentiate between those aspects of the RM process which were generic to all product lines and those which were product or product line specific. The key benefits of a generic RMS template are that it (1) provides some level of harmonization for RM processes (e.g., concepts), (2) allows for improved interoperability between RM artefacts, (3) supports the maintenance and further development of RMS as there is only one template, (4) facilitates product documentation and user support (i.e., documentation and support resources need not be distributed over multiple RMS designs, making the learning and adoption of RMS artefacts easier and faster), and (5) makes it possible to outsource the RMS development, maintenance and user support to external service providers if the internal IT organization cannot or does not want to take long-term responsibility for RMS. This need for a productized RMS solution is one of the reasons why we do not describe the BUs or the first RMS designs in detail: instead, we highlight the meta-requirements, which applied in all the BUs and shaped the development of a generic RMS template.

While the RMS artefacts were implemented on top of Lotus Notes™, this paper aims at providing guidance for implementing RMS artefacts also on other platforms with capabilities for enacting role-oriented workflow processes. Thus, we do not discuss the particular features of Lotus Notes™, since such information is readily available from vendors and quickly outdated, too. It suffices to note that Notes™ provides appealing features such as a distributed database architecture[3], stringent access controls, multi-platform support, threaded discussions and possibilities for mobile access.

B. Theory Development for Designing Requirements Management Systems

Theoretical guidance is relatively scarcely available in the literature to help build RMS.

[3] That is, identical copies of the RMS database were maintained on several servers.

This paper draws upon relevant literature and the empirical research at Nokia to start building an ISDT for RMS in NPD organizations. In the following, we characterize design theories and present definitions of the components of an ISDT. Subsequent sections elaborate on these components to develop the preliminary ISDT for RMS.

Design theories have several distinct characteristics that differentiate them from other theories [52, p. 40-41]:

1. They must deal with goals as contingencies. For example, the ISDT for RMS states that if the business goals (shortening the cycle time of NPD while improving product quality and maintaining or reducing the total resources required) are to be achieved, then RMS should be designed and used to redesign RM processes.
2. They prescribe both the properties an artefact should have if it is to achieve certain goals and the method(s) of artefact construction.
3. A design theory can never involve pure prediction or explanation. For example, the ISDT for RMS explains what properties RMS should have and how RMS should be built and predicts that an RMS will achieve its goals (i.e., supporting the attainment of the business goals) to the extent that it possesses the properties and is built using the methods prescribed by the theory.
4. They are prescriptive, composite theories integrating explanatory, predictive, and normative kernel theories from natural and social sciences and mathematics into design paths that realize more effective design and use. They involve both the application of scientific theory to design artefacts and the use of the scientific method to test design theories (usually by building and testing the artefacts empirically).
5. Design theories tell "how to (achieve the goal)/because" whereas explanatory theories tell "what is", predictive theories tell "what will be", and normative theories tell "what should be (the goal)".

For example, a theory about the role of RMS champions might be devised (1) stating that NPD organizations aim at improving communication, coordination, and collaboration by means of leveraging RMS and (2) hypothesizing that NPD organizations with powerful RMS champions are more likely to achieve this goal. The purpose of the theory is not to achieve the goal but to predict that goal achievement is more likely when a certain condition (powerful RMS champion) is met. Such theory development can only contribute to the foundation laid by this paper, if it is done in the context and as a part of developing a design theory. For example, the theory can be used to develop those aspects of the design theory that guide the design process of RMS together with the design of RM processes.

Walls et al. [52, p. 42] argue that an ISDT must have two aspects: "one dealing with the product and one dealing with the process of design." They define four components of the product aspect:

(1) *Meta-requirements* describe the class of goals to which the theory applies. Meta-requirements for building an ISDT for RMS are described in Section III.

(2) *Meta-design* describes a class of artefacts hypothesized to meet the meta-requirements. The meta-design of RMS is presented in Section IV.

(3) *Kernel theories* [TKK1]are theories from natural or social sciences and mathematics governing design requirements. They are discussed in Section VI.

(4) *Testable design product hypotheses* are used to test whether the meta-design satisfies the meta-requirements. They are discussed in Section VI.

Walls et al. [52, p. 43] define three components of the process aspect:

(1) *Design method* describes procedures for artefact construction.

(2) *Kernel theories* of the design process aspect [TKK2]are theories from "natural or social sciences governing design process itself" [52, p. 43].

(3) *Testable design process hypotheses* are used "to verify whether the design method results in an artefact which is consistent with the meta-design" [52, p. 43].

These three components are discussed in Section VI.

III. Meta-requirements for Groupware-Based RM processes

In this section, we present the meta-requirements, which shaped the implementation of RMS at Nokia, and introduce a framework for analyzing them. These meta-requirements were derived from a literature survey and semistructured interviews with middle-level R&D managers at Nokia BUs. The interviews typically lasted about two hours and were conducted to identify (1) what objectives should be placed on RM support and (2) what factors should be accounted for as potential pitfalls in RMS deployment. The results of most interviews were quite similar and in line with earlier reports in the research literature, which made it easier to synthesize them to support RMS development.

The framework (see Table 1) considers meta-requirements in relation to the generic functionalities that RMS has to offer in support of (1) communication, (2) control and (3) change. Here, *communication* is understood as the ability of RMS to disseminate requirements information within the NPD organization, including information about the rationale for RM and its relationships to external environment, other NPD processes and attendant organizational responsibilities, among other things. *Control* is needed to ensure that requirements are dealt with in accordance with approved principles and procedures and in alignment with the objectives that have been placed on RM. Because products, technologies, and customers *change*, it is crucial that RMS artefacts remain amenable to adjustments at all levels of RM activity.

Each of these functionalities can be examined in relation to (A) context, (B) process and (C) content (see Table 1) characterizing the domains where RMS artefacts are used. Here, *context* refers to those attributes of the external environment (e.g., market segments), technological positioning (e.g., standards, interfaces to other products) and organizational setting (e.g., allocation of responsibilities) that are needed to establish a sufficiently encompassing framework for storing, refining, assessing and retrieving requirements. *Process*, in turn, refers to the temporal structuring, sequencing and monitoring of the tasks that are carried out in order to derive approved product specifications from requirements information. Finally, *content* refers to the means of describing requirements (e.g., notational conventions).

Table 1. Framework for Analyzing Meta-Requirements for RMS

Table 1 lists several meta-requirements, which have been derived from an examination of the generic functionalities. We conjecture that these requirements apply also to other large and multi-site NPD organizations which seek to develop consecutive releases of high-technology products within tight schedule constraints; this conjecture appears warranted due to the close parallels to the requirements identified by Foster et al. [16], for instance. Below, we discuss these meta-requirements in more detail.

A. Meta-Requirements in Support of Communication

Integration of Functional Units

Both our interviews and earlier accounts in the literature suggested that all stakeholders – product marketing and R&D, in particular – should contribute to the product definition. Griffin and Hauser [18] and Trott [54], for example, report that product success is more likely if marketing, R&D, and production share information on all aspects of the product. Theofanos and Fleeger [42] cite several benefits associated with the deployment of multi-disciplinary competencies in requirements elicitation (e.g., improved system usability, testability, maintainability, and overall quality). Käkölä [48-51] states that the disjuncture between the design and use of products should, if possible, be eliminated by understanding the various domains of use, positioning the products in the domains and value networks during the design stage, and designing the products so that they make this positioning transparent and clear to those using the products in the deployment stage.

In their empirical study, Cooper and Kleinschmidt [11] attribute the higher performance of product development teams to the adoption of cross-functional groups, effective communication, and overlapping product-development phases. Burchill and Fine [8] conclude that increased functional integration contributes to the participation of individuals in *all* early NPD phases and leads to a more credible design. This, in turn, is likely to strengthen the developers' commitment to the product concept, reduce misdirected development efforts and shorten the overall product development time (see also [5]).

Our interviews were very much aligned with these studies. For example, a senior marketing and sales specialist in one BU stated:

> "I believe that the biggest challenge is not in whether the marketing people communicate well or whether the marketing and R&D people integrate well. We usually have a small team of marketing and R&D people that develop a common understanding of the product concept or feature and convince our top management that it should be implemented, but the developers will be people who are not directly involved in that front-end process. The biggest challenge is in communicating the original need and idea to those people that eventually start defining the specification and implementing the solution so that they really understand holistically why this is done and what need it should fulfil. After all, this communication will influence to what extent the solution will fulfil the original need and ideas."

Because these processes involve stakeholders from several functions, it follows that methods and tools for the earliest phases of RM processes should be relatively simple. This is because there are several interfaces to organizations (e.g., sales, product marketing) that come into contact with requirements only intermittently and are not keen on learning complicated processes, methods, or tools.

The solicitation of requirements is crucial, wherefore the submission of new requirements can be explicitly encouraged by asking the stakeholders for their input periodically or possibly by giving rewards for the best ideas, for instance. As a practical design step for ensuring that the expertise of different stakeholders can be fully leveraged, it may be advantageous to send automated requests through escalation procedures or automatic e-mail notifications (see, e.g., [17]). These mechanisms relieve the stakeholders from the need to continually monitor the RM process. Conversely, putting excessive demands on the initial requirements, for instance by insisting on detailed estimates about the impact[4] of each requirement, may discourage stakeholders.

Elaboration and Application of Domain Models

To support access to requirements information, it is crucial that sufficiently rich domain models are applied when organising requirements. These domain models need to reflect the diversity of data and, among other things, they should describe the physical and logical attributes of product structure (e.g., performance and reliability). In addition, they should cover the organizational entities that are responsible for the management of requirements information, whereby the RMS artefacts can assist in the allocation of authority and responsibility.

Traceability

Traceability support is motivated by several reasons. First, traceability enhances the level of communication support, as the originators of requirements can identify the documents that their input has lead to and, conversely, the designers can get in touch with originators of the requirements. Second, traceability contributes to a more legitimate and open decision process by enhancing cross-functional integration and broad participation [44]. Third, a traceability program is called for whenever the project is complex, the safety of the system is critical, or there is a need to see different types of documents and their interrelationships [45]. For example, initially submitted requirements vary greatly (i.e., functionality, required work effort, number of interfaces), and thus there is a need to support the (1) the separation of large requirements into smaller constituents and (2) the aggregation of smaller requirements into larger entities. In addition, because in large organization several persons may produce similar or related ideas, there is a need to support the linking of new ideas to earlier ones and, more generally, the creation and maintenance of links between different layers of documentation (e.g., new ideas vs. derived feature proposals).

[4] The impact of a requirement refers to the implications that its implementation would have on the product architecture, release scheduling and the allocation of work between development teams, among other things.

Accessible IT Infrastructure

In large, multi-site NPD organizations, widespread access to a robust IT infrastructure is a prerequisite for the deployment of groupware support for RM processes. Furthermore, RM processes have many interfaces to other processes (e.g., sales, customer services) and to other organizations (e.g., customers and subcontractors), which in turn poses challenges for systems interoperability. In practice, this implies that proprietary RMS platforms cannot be considered for the user-interface front-ends of RMS artefacts unless they support Internet standards and protocols. For example, customer service portals can be built and requirements management support can be one of the services provided, when the RMS platform is compatible with Internet and thus accessible through standard web browsers.

The geographical dispersion of NPD sites is a critical consideration in evaluating the database architecture of the RMS platform. Moreover, the platform must integrate well with other components of the corporate IT infrastructure (e.g., email, modelling and testing systems, spreadsheet-based analysis tools) and be deployable cost-effectively within the entire NPD organization.

Overall, the choice of an RMS platform is a strategic decision with regard to the possibilities that the interoperability of different IT systems and their evolution may open up.

B. Meta-Requirements in Support of Control

Elaboration of Control Procedures

Having the requirements in an RMS increases their visibility and thus makes it easier to control them. Nevertheless, explicit control procedures are needed to guarantee that requirements evolve only in intended and controlled ways. Such procedures also facilitate the dissemination of domain knowledge: as change requests are handled in the context of the documents they pertain to, newly hired R&D engineers can better understand the reasons behind change requests and the implications they would lead to. Thus, instead of merely recording the outcome of a change request assessment (accept/reject), it is instructive to provide detailed information about the expected impacts caused by the implementation of these requests (e.g., schedule, work effort).

Implementation of Incentive Structures and Clear Allocation of Responsibilities

People collaborate only if they have an incentive to do so [1,33-34], either due to social (e.g., recognition of expertise) or managerial (e.g., monetary) incentives. To some extent, groupware can assist in the implementation of incentives because it makes the content and extent of computer-mediated collaboration visible and measurable. Visibility, in particular, leads to a wider awareness of the different types expertise in the NPD organization and induces self-control over incorrect information [1, p. 273]. Measurement (e.g., number of documents submitted to the RM process), on the other hand, can be used as a basis of managerial incentives.

An important dimension of control is the ability to determine the originator of any document as well as the persons who have made changes to it [4]. This contributes to (1) the clarification of responsibilities, (2) increased visibility of knowledge (i.e., who knows what), and (3) the implementation of incentives. Specifically, responsibilities can be allocated by associating with each document an *owner* who has relevant expertise and is held accountable for that document.

Support for Decision Making

Critical decision milestones in NPD include (1) go/no-go decisions about which feature proposals, derived from submitted requirements, will be subjected to in-depth studies and, once such feasibility studies have been completed, (2) the selection of which features will be transferred to development work [8]. These decisions are taken within the framework of product strategy, which builds on technological roadmaps, and analyses of market segments. In the above decisions, product strategy can be accounted for by explicitly defining the criteria against which the proposed product features are assessed.

To assist in organizational decision making, RMS should support (1) the identification of stakeholders who - in view of their experience and responsibilities - are positioned to assess suggested product functionalities, (2) the solicitation of evaluation and review reports on feature proposals, as well as informal feedback that contributes to a better understanding of the product and its mission (e.g., changes in the customer's competive position); and (3) the maintenance and use of relevant document attributes (e.g., document's author, time of creation, relationships to other documents). A key determinant of decision quality is how well knowledge of the NPD organization as a whole and particularly the expertise of its more experienced members is leveraged. Here, an important design concern is the facilitation of mutual critiquing to enhance the quality of information generated during requirements assessment [4].

C. Meta-Requirements in Support of Change

Responsive to Organizational Changes

Modern NPD organizations, especially high-technology companies, are often subjected to rapid structural changes [9]. In consequence, RM processes and associated groupware support should be flexible so that new structures and responsibilities can be introduced without necessitating major disruptions in RMS use. This implies that the NPD organization needs to maintain information about (1) the context in which the RM processes take place and (2) the roles which the different stakeholders have in these processes.

Specifically, RMS should also permit the realization of alternative subprocesses so that error reports, for instance, can be treated differently from requests for major functional extensions (see for example [2]). Thus, RMS should allow for flexible parameterization instead of enforcing a single "hard-coded" RM process: Smith and Reinertsen [39], for example, note that "without alternative processes, all projects tend to get sent through the same process, a common denominator that suits no objective well."

Scalability

Scalability is critical in industrial projects with thousands of documents and hundreds of daily users. The need for scalability manifests itself at the levels of having (1) a robust IT platform, which is capable of handling the sheer volume of documentation, and (2) appropriate representational schemes with the help of which requirements are organised and retrieved. The first of these items is mainly determined by the underlying IT platform. The second item implies that the approach to metamodeling (e.g., product structure, roles and responsibilities) should, if possible, be flexible enough to accommodate organizational changes and extensions to the RM process itself. Since these extensions may not be known at the time when the RM process is first introduced, RMS should build on generic principles which are likely to support the instantiation of new subprocesses (e.g., handling of errors related to a new interface to another product line).

IV. Meta-Design for the Groupware-Based RMS

This section outlines a generic meta-design for RMS, which responds to the above meta-requirements. This meta-design is by no means the "best" or the only one; rather, it exemplifies the design that was based on the discussions with BU representatives and then evolved in response to the feedback from RMS pilot implementations at Nokia BUs. The meta-design is presented from an informational perspective, while the next section considers organizational and processual dimensions (see, e.g., [35] for a discussion of complementary modelling perspectives).

A. Phases of Requirements Management

Associated with the work of Yeh [46], we are concerned with the following phases of requirements management (see Table 2):

Table 2. Phases in Requirements Management.

It is instructive to associate these five phases with the attendant processes of organizational memory, i.e., acquisition, retention, maintenance, search, and retrieval of information [40,43]. That is, requirements capture and categorization relate, respectively, to the acquisition and retention of information; requirements refinement and assessment relate to the maintenance of information; and finally, requirements follow-up deals with the search and retrieval of information.

Requirements Capture

In this phase, the purpose of RMS is to facilitate the collection of requirements from all the relevant stakeholders (e.g., customers, sales, product marketing, R&D). To foster the acceptance of RMS, it is advisable not to impose undue constraints on the representation and substance of requirements. The key issue is to make the RMS accessible to all stakeholders so that they can easily submit new ideas at the time when they are first recognized.

Requirements Categorization

Submitted requirements must be placed into their proper context to support their retention, retrieval, interpretation, clarification and validation. Here, it is crucial to develop domain models about the customer's use environment ([8], e.g., business processes), the physical and logical product structure, as well as the associated roles and responsibilities within the NPD organization. The design of RMS features for this phase is presented in IV.C.

Requirements Refinement

The raw material produced through requirements capture and categorization needs to be converted into prospective product features about which decisions can be made. This takes place through three processes:

1. *Requirements aggregation*, where small entities are assembled into well-defined product features,
2. *Requirements separation*, where large concepts are divided into smaller ones that can be meaningfully analyzed. In system deliveries, for instance, the separation process includes the allocation of related requirements to different product lines.
3. *Elimination of redundancies*, where related ideas – possibly proposed by several people independently – are linked to each other to avoid duplication of effort in analysis.

RMS support for these processes consists of traceability links and layering of documentation. The design of RMS features for this phase is presented in section IV.B.

Requirements Assessment

In this phase, requirements and proposed product features are evaluated, in order to decide which features are aligned with the product strategy and commercially and technically feasible. These decisions should be based on the best available information, and they need to be taken in a timely, consistent, justifiable and identifiable manner so that, later on, it is possible to determine where, when, and by whom and on what grounds the decisions were taken. RMS support for this phase is presented in section IV.E.

Requirements Follow-Up

Once the product decisions have been made, the features provide a foundation for the later phases of product development. To support these phases (i.e., detailed product specification, implementation and testing), the requirements in the RMS should be made visible to the stakeholders in the NPD organization that take part in them. In particular, additional insights can be gained by allowing the R&D personnel to trace the requirements back to their origins, while those who initially submitted the requirements might be interested in examining just how far the implementation of corresponding features has progressed. Sections IV.B and IV.D discuss support for the follow-up processes.

B. Layering of Documentation

To separate the inputs to the RM process from their implications in terms of new product features, it is helpful to establish a layered documentation structure. This also makes it possible to apply different quality controls to the different layers of documentation.

In its simplest form, the layered documentation structure consisted only of two layers: *product ideas* and *feature proposals* (see Table 3). Conceptually, these terms distinguish between identified customer needs and derived product functionalities; much in the same way as the terms "user requirement" and "system requirement" separate customers' concerns from technical considerations in structured requirements analysis (see, e.g., [41]).

Table 3. Two-Layered Document Structure.

The two-layered documentation structure supports the channelling of new product ideas into suggestions about prospective product features so that go/no-go decisions about the launch of feasibility studies can be taken (c.f., section III.B). Extending the scope of groupware support beyond feasibility studies would call for further documentation layers. Additional documentation layers might also be called for if the product ideas differ considerably in size (e.g., number of interfaces, resource expenditure), whereby intermediate layers would assist in splitting broad concepts into smaller ones.

A further reason for the layered documentation structure is that different customers often have similar requirements while, internally, several people in a large NPD organization may come up with related ideas. As a result, the refinement of every new product idea into a full-fledged feature proposal would lead to a duplicate effort, which can be avoided by linking related product ideas to a single feature proposal. Thus, the creation of links between product ideas and feature proposals is driven by the need to implement an efficient assessment process.

Bi-directional links between product ideas and feature proposals provide full traceability support (c.f., section III.A). Specifically, these links permit the originators of product ideas to examine the feature proposals that their input has lead to, including the status of these proposals in the NPD process. Traceability links are also helpful in that designers can access and study the ideas, which motivated the development of the feature proposal. Thus, one of the advantages of traceability links is that the designers can get in touch with the originators of product ideas.

In the RMS artefacts at Nokia, the number of incoming ideas has been several times (e.g., three to five) larger than that of derived feature proposals, which is indicative of the need to have a layered documentation structure. The links between product ideas and feature proposals often involve many-to-many relationships, which can be maintained by the responsible document owners.

C. Domain Modelling

At the beginning of the RM process, new product ideas are first associated with a domain model about the product structure, which is then used to allocate responsibilities for the further processing of these ideas. This relieves the contributors of new product ideas from the need to attend to the processing of the ideas they have submitted. The allocation of these responsibilities also ensures that the NPD organization can trust the validity of the RMS contents.

For the sake of clarity, it is helpful to divide the requirement documents (both product ideas and feature proposals) into sections, which provide answers to specific questions. Table 4 presents such a division and describes associated attributes.

Table 4. The Generic Structure of a Requirement.

For *product ideas*, the origins of the requirement must be specified. For *feature proposals*, the origins section may be omitted since this information can be obtained from the product ideas through the traceability links. On the other hand, the analysis section of feature proposals must be completed with care.

An RMS instantiation can be developed by choosing sections from the above template, whereby the domain models about the product structure and the NPD organization need to be developed. To manage the workflow, it is also necessary to choose status flags that describe different phases in the processing of requirements (e.g., *New - Categorized - Analyzed - For Review - Approved / Rejected / Postponed),* as well as the persons to whom the requirement should be assigned in these phases, for instance on the basis of the expertise they have.

Role- or Phase-Sensitive Document Layouts

A possibility in context modelling is that of making the document layout contingent on the particular phase that the document is in. For example, the due date for a review report and the names of the reviewers might be placed at the top of a document that is currently under review so that these pieces of information are easily noticed.

However, our experiences at Nokia suggest that context-sensitivity needs to be employed with caution. First, the development of context-sensitive layouts may lead to a more complex process and complicate the design of the RMS, making it harder to adapt it to future changes. Second, context-sensitivity must be implemented intuitively, for otherwise those users who do not use the RMS regularly may find difficulty in understanding "what is going on" or why the system behaves the way it does.

History

A critical element in RM (e.g., in terms of traceability; see section III.A) and control of information (section III.B) is the development of an organizational memory of how, when and by whom the requirements have been modified. Towards this end, the requirements should be appended with a history section which accumulates triplets consisting of (1) the date of modification, (2) the name of the person who modified the document, and (3) the status in which the document was as a result of the modification (e.g., *New*, *Rejected*, *Approved*). By examining these history sections, it is possible to examine how the requirements have evolved and, if necessary, contact the persons who made the changes. Such a possibility is especially useful when a routine process breaks down unexpectedly and the reasons for the breakdown must be found and eliminated to continue the execution of routines (c.f., [48, 50]).

D. Information Navigation

In Lotus Notes™, information search and retrieval are supported through views. These are hierarchical document lists in which one or more of the leftmost columns are sorted with regard to the corresponding document attributes. Combined with the ability to store multiple types of information, views provide a powerful support for organizational memory ([28], pp. 263-264).

To make the most out of views and the associated possibilities for navigation, it is helpful to exploit the different dimensions along which requirements related information might be searched for. At the broadest level of analysis, it is possible to identify role-, content-, and process-oriented views, of which examples are given in Table 5.

Table 5. Role-, Content-, and Process-Oriented Views for Information Navigation.

It may be desirable to restrict the visibility of some requirements to a subset of the NPD organization. Such restrictions reduce the possibility of information leaks and decrease the risk of premature conclusions about the features, which might be forthcoming in future, product releases. To achieve this, read and write access rights can be managed at the level of individual documents. At Nokia, the document owners were responsible for determining the persons (or groups thereof) with appropriate rights for read and write access.

In addition to the views in Table 4, two other RMS features were helpful in supporting information navigation: *To Do*-views and e-mail notifications. *To Do*-views contained, from the perspective of each user, only those documents that were specifically under his or her responsibility, or alternatively, assigned to him or her for specific action (e.g., review report). With these views, the users could readily complete all the tasks for which they were responsible. Furthermore, e-mail notifications were sent to the persons from whom additional inputs and reactions were solicited. These notifications were instrumental in gaining the attention of those experts from whom inputs were being solicited on an occasional basis (c.f., section III.A).

E. Support for Requirements Assessment

As discussed in section III.B, a central issue in RMS design is the question of what tools should be provided to support the assessment of prospective product features. Formal evaluation approaches such as quality function deployment [3, 20, 22] lead to a more thorough analysis as they force the NPD organization to systematically examine proposed features in view of the objectives that have been placed on product development at large. Yet, the usefulness of complex evaluation practices may be limited in distributed NPD organizations: the sheer amount of submitted requirements (say, dozens of new product ideas per a day) limits possibilities for employing lengthy evaluation procedures which would call for inputs from several experts. Thus, more "lightweight approaches" to decision support may be preferred.

In our case at Nokia, formal evaluation practices were not integrated into RMS. Instead, the most significant evaluation concerns for feature proposals were identified and itemized on separate review forms (e.g., usability, market value, and technical feasibility). For each proposal, the reviewers employed these forms to supply their decision recommendation (e.g., accept or reject) and a brief justification for it. This approach was chosen as compromise between (1) the complexity of the method, (2) the amount of information needed in support of product decisions, and (3) the work effort which could be devoted to the analysis.

More generally, our experience suggests that it may be sufficient to raise the NPD organization's awareness of the relevant decision criteria without enforcing formal evaluations along them all. First, the burden and costs of obtaining formal evaluation reports from the NPD organization should not exceed the benefits, which are perceived to result from such evaluations. Second, the role, applicability, and importance of the criteria vary over time, and they depend on hard-to-quantify measures such as the strategic role of a certain customer in long-term partnerships. Third, while formal approaches (e.g., voting, scoring, and weighting) are helpful in the aggregation of assessments, their straightforward application may nevertheless be too simplistic and mechanistic, especially if applied to features that are image-laden or otherwise hard to quantify (see, e.g., [23]).

To support the assessment of feature proposals, RMS was coupled with an organizational domain model that explicated the fields of expertise of the different persons in the NPD organization. This was done by implementing adjunct databases, which contained information about the persons who were specialists in, for example, specific markets or standards. With the help of this information, feature proposals could more easily be allocated to the most competent persons for review. This modelling of "know-who" also contributed to the visibility of domain knowledge (section III.A) and the adaptability of RMS artefacts in response to organizational changes.

F. Allocation of Responsibilities

In RMS, requirements control was achieved by assigning a responsible owner to each product idea and feature proposal. This owner was held accountable for ensuring that (1) the documents assigned to him or her were translated into more detailed representations in preparation of product decisions and that (2) the evaluation and review rounds were undertaken in due time. Furthermore, the owner was responsible for monitoring changes or feedback to these requirements, and for the introduction of sanctions, if necessary.

Explicit change control procedures were enforced especially in cases where other deliverables (design specifications, software code, test plans, customer documentation) had been derived from the feature proposals. These controls were, again, enforced through organizational responsibilities so that the owner was, for instance, held accountable for ensuring that the acceptance of a change request would not lead to adverse side effects or unforeseen impacts (see for example [24]). Here, traceability support and domain models about external interfaces were instrumental in the implementation of change controls.

G. Manageability through Metrics

An appealing aspect of groupware-based RM processes is that the infrastructure can automatically provide some quantitative metrics. These metrics provide a clearer view of how the NPD process is functioning and what shortcomings stand in the way of better performance. Thus, the metrics not only highlight what is happening but also suggest areas for process improvement [14,38]. A reasonably holistic measurement program is, in fact, a prerequisite for understanding the NPD process [36].

From the perspective of groupware implementation, the easiest metrics to obtain are those which deal with (1) the volume of documents in a certain category of a domain model, (2) the duration that documents spend in the different phases of their lifecycle, and (3) combinations of these. For instance, a metric about the number of documents submitted by a given author is an example of the first category, while the percentage of documents that have not been processed within the targeted review times belongs to the third category. Table 6 lists metrics that were provided through combinations of metrics in the above categories.

Table 6. Examples of Metrics.

Although groupware offers extensive possibilities for obtaining different types of metrics, these metrics should nevertheless be derived from the overall objectives placed on the NPD process. Here, approaches such as goal-question-metric approaches (see for example [6]) are helpful in clarifying the process goals and translating these goals into specific metrics. For example, if the main target in improving NPD processes were that of shortening the cycle time, then the role of time-oriented metrics would be highlighted, perhaps at the expense of effort-related metrics.

The metrics in Table 6 are based on information that can be (1) supplied by software agents running on the RMS database or (2) derived from logs about database usage. Apart from these, an RMS artefact can be used to request subjective estimates for the computation of other metrics such as how long the users work with certain documents. While there are few inherent limitations on collecting such subjective metrics, the cost and effort involved must nevertheless be weighed against the benefits of acquiring them [13].

V. Validating the Meta-Design of RMS

In order to assess the validity of the meta-design of RMS, we report experiences from RMS support in Nokia based on interviews with NPD process owners, product managers and other RMS users in the product lines, which had adopted RMS artefacts by the end of 1990s.

A. The Adoption of Revised RM Processes

Before the adoption of RMS artefacts, the RM processes were relatively informal since much of the communication took place through e-mail messaging and phone calls. With the rapid growth of the NPD organization, however, it became increasingly difficult to gain an overview of the RM activities because these types of communication did not leave traces that could be easily followed.

During the RMS deployment, an RMS representative – typically the owner of the RM process – was appointed at each BU. The task of this representative – or RMS champion – was to motivate the adoption of the RMS artefact, to organise training sessions and, more generally, to answer questions on the artefact and its use. A benefit of having an appointed representative was that the RMS users could address questions to a person they already knew in their own NPD organization. As a result, they felt that the artefact was more of their own, rather than a solution that was being imposed by other Nokia units.

While the stakeholders acknowledged the communication problems before the field trials were started, it was nevertheless felt that an intensive internal training program would be needed to describe the objectives of RMS deployment. Another reason for the training program was that the users would have to adopt the artefact at the same time: since the purpose of RMS was to support communication, coordination, and collaboration, it would be successful only if all the targeted users would start using it.

In the training, the RM process was described in terms of (1) how requirements evolve through consecutive phases and (2) what actions and inputs are requested from the different stakeholders during these phases. Thus, an attempt was made to first describe the RM process *as a whole* and the role that the different RMS users had in it. This was because several users reported that it was easier to learn details (such as features of the user-interface) once they had understood how the RMS artefact allowed them to contribute to the RM process. Figure 1 gives an example of a schematic diagram that was used in presenting the RM process.

Figure 1. A Process Illustration.

Since the activities during the earliest phases of NPD had been relatively informal, the BUs were adaptive in terms of accepting a more formally defined RM process. In fact, some interviewees from R&D noted that it is probably easier to adopt a completely new process than to revise an existing one for which there are deeply rooted behavioural patterns in the NPD organization. This may be one of the reasons for why the simple RM process design, based on the two-layered documentation structure in section IV, did not meet much resistance in the BUs.

From the viewpoint of scalability, it was critical to construct domain models with the help of which the amount of requirement information could be managed. A potential pitfall, encountered in one of the BUs, was that the need for such domain models might not be evident when the amount of documents is still small. Yet, as the amount of requirement information grows, the retrieval of requirements becomes increasingly difficult without proper domain models. The consideration of domain models can also be motivated by persistency: once a domain model has been in use for some time, one would have to analyze and possibly revise much of RMS contents in order to replace the domain model by another one.

B. Implications of RMS Deployment on RM Processes

The introduction of the RMS artefact clearly increased the visibility of requirements information and defined a traceable RM process. A marketing specialist stated:

> "Marketing people are located all over the world and they do not often know very well the organization and who is responsible for what. I believe that, for example, a person from Hong Kong, who stays here [in Finland] for a month in the beginning [for training] and then returns to Hong Kong, will not understand very well where to send the requirements. RMS solves such problems because the person simply needs to input the requirements and select the related areas [in product structure] from a list. Immediately when somebody re-

sponsible for the area has started working on the requirement, one can see the name of that person and know that the matter is being dealt with."

In fact, once the artefact had been in use for some time, some users in R&D were surprised by the amount of information that was being shared. This suggests that before it was adopted, much of this information was sent through one-to-one or one-to-few communication channels (e.g., e-mail), which were not visible to the rest of the NPD organization. After the artefact was deployed, the NPD organization as a whole, then, became more aware of how new product ideas and feature proposals contributed to the development of subsequent product releases.

Because the RM process had not been formally defined before the artefact was adopted, it is difficult to ascertain to what extent the artefact may have sped up the development of new feature proposals. Some users also observed that, in the end, development speed or even the number of new feature proposals created in a certain time frame should not be primary objectives: rather, the artefact should ensure that the *right* features are developed, i.e., it should help in acquiring all the relevant requirements to support the definition of new features. In this respect, it was perceived to be successful: for example, the ability to access it through mobile connections was cited as a feature, which facilitated the submission of new requirements.

Experiences with RM support also suggest that RMS has helped in improving the quality of the feature proposals. This was because the artefact (1) assisted in obtaining a broader knowledge basis from which the proposed features were derived and (2) enforced systematic review rounds through which critical reviews were brought in from several experts. Taken together, the result was an RM process that produced more candidates, which were more carefully assessed in the NPD organization.

The artefact also seems to have clarified the roles, which the different stakeholders had in the RM process. That is, while users reported that they had become more aware of the RM activities, they also mentioned that they had a better understanding of the ways in which they can best contribute to the RM process. Here, the development and maintenance of adequate domain models – which extend to organizational responsibilities – seems critical: these domain models not only help in the identification of specific types of expertise (e.g., "who knows about such-and-such a standard?"), but they also allow the development of metrics based on particular areas of responsibility (e.g., "how long does it take to obtain feedback on such-and –such a standard?).

Many of Nokia's BUs have R&D sites in different countries and even on several continents. Frequently, these sites need to collaborate on joint projects, and here the artefact has become helpful as it provides a shared communication platform, which contributes to the integration and co-ordination of RM activities at the different sites. It also supports benchmarking and organizational learning across product lines and sites by automatically collecting performance data based on shared and agreed upon metrics (c.f., [50-51]). A further benefit of having a shared RMS is that those users who move to another site can continue to work without much interruption because there is little need to learn site-specific RM practices or solutions.

In many product lines of a company that has grown rapidly since early 1990s and has had to focus carefully on hiring and managing people and developing their competencies, RMS has defined an organizational memory which allows new employees to learn about how the NPD organization has dealt with the earlier product ideas and feature proposals. Thus, the RMS can also be seen as a training tool with the help of which new employees can more quickly become productive members of the NPD organization. While the value of such on-line training is difficult to estimate, some of our interviewees noted that it has been significant.

C. Challenges in Reaching Institutionalized RMS Deployment

While the early adoption of RMS was almost trivial whenever there was a strong RMS champion, it was often extremely challenging and time-consuming to reach the stage of institutionalized use and the associated benefits described in Section V.B. For example, a marketing and sales specialist responsible for channelling customer requirements to product managers in a few product lines stated several months after the adoption of RMS:

> "I use RMS very seldom. I just recently checked whether something has happened there, but not a lot, it has been quiet there. But there has not been an e-mail note either stating that use that [RMS artefact] again … you start to assume … that unless there is some command about using a system, nobody else is going to use it anyway."

Most reasons for the challenges were of business, organizational, and processual nature, but some were also technical and RMS related.

The most notable challenge was that often requirements were not entered in the artefacts in the first place. Business was very hectic and business customers often casually suggested new functionalities to sales people amidst meetings or negotiations on some other issues. If the customers did not actively pursue their ideas, it could be difficult for sales people to ascertain whether the customers really wanted what they had casually asked for. Because the RM processes were undefined in the beginning and only gradually constructed with the help of RMS artefacts, strict rules for entering such requirements in the artefacts were missing. As a result, such requirements were not necessarily entered. On the other hand, requirements critical for closing business-to-business deals quickly with customers were not always entered either. Negotiators could bypass RMS artefacts entirely and seek immediate top management approval for new features if they considered the business benefits to justify such actions. Our interviews did not reveal how common such activity was but the interviewees stated that bypassing the artefacts was mainly done when it was absolutely necessary for fast response management. For example, a senior marketing and sales specialist in one BU stated:

> "Requirements that are surfaced during business negotiations and the decisions related to them do not always end up in our RMS. Such a requirement may already exist [in the RMS artefact] but it may not exist as well and then the approval of management can be sought even through direct phone calls. It is a business issue that in that stage we can make a positive decision and commit to implementing this and then a separate exercise is needed to find out how to fit the implementation in the [product development] schedule and so on. This can be done without the artefact, which of course is not necessarily good from the view point of the [NPD organization as a] whole but from the view point of our business it can be necessary."

The desire to use phone and other media besides the RMS artefacts was understandable especially in the beginning of RMS-enabled RM process development. Business customers and marketing people were sometimes alienated because they typically wanted to know the statuses of the requirements they had entered (i.e., whether somebody in R&D was working on the requirement and so on) but often the statuses were not updated quickly enough. At the time of the adoption of RMS, the growth of the business and the organization was enormous resulting in an ever-increasing number of requirements to deal with. Development of the RM processes in this context took time and effort. Moreover, the R&D people typically had a backlog of requirements to work with that reduced their available resources for dealing with process development and new requirements. The business people needed feedback in their daily routines quickly and did not see the point in entering requirements in the artefact if there did not seem to be fast response. For example, a marketing and sales specialist stated:

> "I want some indication whenever something is happening [to a requirement I have entered]. There should be some defined processing time in the process

> and the tool so you would have some deadline by which you could expect an answer. What is particularly unmotivating in requirements management is that you do not get any feedback. If you put something there, it will then just stay there."

Marketing people could also be left in the dark in the later stages of product development even if the R&D people actively worked on the respective requirements. The key reason for this was that RMS supported product development work only until the product features for the next release were frozen. In some product lines, the access rights to the RMS artefact of a product development project were even granted only to those responsible for creating the requirements specification of the project. Designers, implementers, and testers of the features used numerous other production tools in the later stages of product development but not the RMS artefact because it did not benefit them directly. Most tools were not integrated with the artefact, so automatic status updates of requirement documents were not possible either. As a result, customers and marketing people could usually trace their requirements only till the end of the requirements specification stage to see whether a particular requirement had been included in a feature proposal or not, and if it had been included, whether that proposal had been accepted or rejected. They could not trace through the artefact what happened to the accepted proposal in the later stages of product development.

Another reason for the limited use of RMS artefacts during the later stages of product development was that product lines and BUs emphasized the role of requirements specification documents as a primary input for the later stages. Designers, implementers, and testers had learnt to rely on the specifications in their work already before the adoption of RMS. They felt that all the necessary information had to be available in the specifications and did not consider the browsing of RMS databases very important. Even a new feature of RMS had to be developed to automate the creation of specification documents to the extent possible. The feature searched from the database of an RMS artefact for the feature proposals accepted for the next product release and exported that information into the specification that could be edited by using a word processor. This specification-centred view of requirements- and product concept-related organizational memory guaranteed high usability and accessibility of limited aspects of the memory but was simplistic and reductionistic as well. For example, most traceability related information could not be incorporated in specifications.

Collaboration between marketing and product development was also hampered because RMS and RMS requirements databases were not shared adequately across product lines and BUs, which resulted in coordination problems. An R&D specialist responsible for requirements management in a product line stated:

> "For example, customer requirements are collected much too redundantly. Projects often collect the same things from the same customers. Whenever projects start, they send questionnaires to the customer interface, that is, sales asking what customer requirements there might be for such and such a product. The customer interface will get so many inquiries that they either do not care about those or respond very infrequently or give answers that do not have much relevance."

Especially the marketing people considered understanding of the domains of use of customers absolutely critical for successful product development. RMS was thus designed to support various representations of the domains of use of customers. For example, business process models of telecommunications network operators could easily be attached to requirement documents. Yet, the interviewees were not convinced of the potential of RMS in this respect. A senior marketing and sales specialist stated:

> "Understanding of the domain of use cannot be conveyed through the RMS artefact. It is a very difficult thing to communicate through any literary way. Maybe the best way to facilitate the understanding of the domain and experiences of users is to provide people with an opportunity to meet the users and hear direct comments from them concerning the products in use or have direct feedback from them concerning new projects and their specifications. For example, busloads of mechanical engineers have been sent to see in what kind of environments those devices will be used and which practical problems exist. It should probably be done more systematically but of course it will be away from product development time so there has to be some sort of a compromise."

None of the interviewees suggested that the challenges in reaching institutionalized use of RMS artefacts could be attributed to flaws in the design of RMS. The underlying reasons for the challenges were organizational in nature. For example, domains of RMS use could have been represented more adequately in the artefact and the status information of requirements in the artefact could have been updated more effectively in later stages of product development, if more resources would have been allocated to the RM processes. An R&D specialist stated:

> "… The RM process has not been thought of a lot. … Organizational structures tend to be such that no people can be found in them that would work full time in the RM stage. They [organizational structures] have been built so there is a

> project organization that at some stage starts to work on the requirements. Even the projects are in practise established when the RM stage has been completed. ... We should have full time people to manage RMS databases and do RM work in the front end and then also to work during the projects to update the RMS databases so we would know for each requirement that now this project has completed design, implementation, and/or testing of the requirement, or the projects should be made responsible for updating the status information in the databases."

VI. Conclusions and Future Research

Our research has been driven by the need of large high-tech companies to accelerate NPD while increasing quality and maintaining or reducing the resource requirements for NPD. The RMS-enabled RM process development has been the means of meeting these goals. In this paper, we have started to build an information systems design theory for RMS. Section III described meta-requirements and Section IV presented the meta-design of RMS, which describes the class of RMS artefacts meeting the meta-requirements. These product-related components of the ISDT for RMS were validated in Section V by empirically analyzing instantiations of RMS in the multi-site product-lines of Nokia.

The RMS meta-design has been deployed to a diverse set of product lines at Nokia. Thus, we believe that it is relevant to other NPD organizations as well. However, a limitation in generalizing the meta-requirements and the meta-design is that they have been shaped by the demands at Nokia's BUs. Thus, we may have emphasized some aspects of RM processes while downplaying others that are relevant in other NPD settings. For example, groupware support for synchronous remote interactions may be helpful in some RM processes but we have not addressed this issue, partly because our groupware platform did not support such interactions. Future research in other NPD organizations is thus needed to refine and validate RMS.

It is beyond the scope of this paper to address in detail the other components of the ISDT for RMS. In the following, we briefly analyze the kernel theories and testable design product hypotheses of the product aspect of an ISDT for RMS. Then we discuss the three components of the process aspect of the ISDT: design method, kernel theories, and testable design process hypotheses. Throughout the analysis, we identify topics for future research and critically review the ISDT approach.

Kernel theories are theories from natural or social sciences and mathematics governing design requirements. Most meta-requirements for RMS (e.g., supporting bi-directional traceability) are derived from RM processes. Some requirements are derived from product domains. Products of large distributed organizations are typically systems products that consist of software and hardware components. Therefore, kernel theories for RMS should integrate systems and software engineering, place RM processes in a holistic product development context, help RMS designers understand this context, and build a common language to communicate with RM specialists and other stakeholders involved with RM (e.g., testing). Process assessment and improvement standards and frameworks such as CMM-I [53] meet these requirements for kernel theories. They are relevant because they have been abstracted from the real-life practices of numerous organizations. They help build a common language because they integrate various supporting theories or management methods such as benchmarking, organizational learning, and Total Quality Management. However, their wide scope is also a weakness from the viewpoint of RMS. They need to be complemented with more focused analyses of the earliest phases of NPD (e.g., [55, pp. 154-177]). In addition, we have identified the organizational memory construct to be useful for designing RMS. In sum, we have found theories drawing on social sciences to play the most important role in the context of RMS. Future research must analyze these and other kernel theories in depth to find out whether they pose new meta-requirements that could lead to the improvement of the ISDT for RMS.

Testable design product hypotheses are used to test whether the meta-design satisfies the meta-requirements. For example, the following hypothesis can be stated: RMS artefacts will (A) increase the depth of analysis of requirements (quality item); (B) improve the traceability of requirements (quality item); (C) reduce the cycle-time of new products (time-related item); (D) increase the number or requirements RM specialists can analyze in a given time (resource-related item). Further empirical research should study these and similar hypotheses in many different types of NPD organizations to refine, justify, and validate the ISDT so that it meets the business goals set for the theory. Addressing these issues would not only help NPD organizations improve their RM processes but also extend the NPD literature towards a deeper appreciation of modern communication technologies.

Design method describes procedures for artefact construction. In the beginning of RMS design, the pilots were kept simple and robust and delivered to domain experts in the product lines in rapid cycles for fast feedback (c.f., [56]). During each cycle, most effort was spent on refining the design concepts. Coding and testing were relatively straightforward activities. Participative design improved quality of RMS, built organizational commitment to using RMS releases, and enabled product lines to gradually (re)structure their RM processes around the rules and resources afforded by successive RMS releases. Participative design could be relied on easily because (A) many experts were IT-skilled engineers who, besides their normal duties, even sometimes built their own groupware applications and (B) most marketing people were very competent in leveraging the corporate groupware infrastructure. This approach resulted in divergent pilots that met local needs well but were not necessarily suited for corporate-wide use. Productization of RMS was then conducted to ensure convergence. After the productization stage had been completed, the baseline design was placed under rigorous change management so new features could be implemented without jeopardizing corporate-wide use of RMS. Our method balanced divergence and convergence stages well and we have found it to work also in other settings and projects. However, the method is abstract and not prescriptive. For example, participative design should be conducted differently when the organizational IT-related competencies are not as excellent as they were at Nokia. Therefore, the prescriptive design method component must be subjected to further research, refinement, and validation.

Kernel theories of the design process aspect are theories from "natural or social sciences governing design process itself" [52, p. 43]. Most IS development methods such as prototyping do not have underlying kernel theories. We did not apply kernel theories specifically to the design process because Käkölä [49] found that the separation between kernel theories of design product and process is somewhat artificial. He thus called for IS design *and use* theories. Structural properties of RMS cannot be separated from the design processes that constitute the properties and the use processes that are constituted by and reconstitute the properties. The kernel theories of product and process aspects of RMS need not be the same but they must be significantly overlapping so that the design theory can be drawn upon to discuss, understand, and improve design and use processes using (to the extent possible) the same terms and concepts [49, p. 117]. Otherwise, the coordinated, recursive development of RM processes and RMS may be hampered and the ISDT may not help organizations achieve their business goals. Future research is needed to identify and further develop such kernel theories for RMS.

Testable design process hypotheses are used "to verify whether the design method results in an artefact which is consistent with the meta-design" [52, p. 43]. We have found no single design procedure to be vitally important in realizing RMS. From a technological perspective, an RMS artefact satisfying the meta-design can be implemented easily and in many ways. Providing an exhaustive list of design process hypotheses is thus outside the scope of this research. Organizational implementation of RMS to achieve the business goals is much more difficult. From this perspective, the most important aspect of the design method is its ability to ensure that RM processes and RMS artefacts are successfully aligned and integrated. Our design method worked relatively well in this respect. Future research can thus experiment with the following and many other design process hypotheses: The design of RMS artefacts must satisfy both (A) local requirements of business units to the maximum extent so that the unit-level buy-in is facilitated and (B) the meta-requirements and meta-design so that a corporate-wide, productized RMS solution can be obtained.

VII. Acknowledgements

The authors wish to thank all the interviewees and especially Juhani Miettunen, Osmo Vikman, Jyrki Heikkinen, Anne Lohva, Timo Kinnunen, Janne Ropponen, Ilkka Tuomi, Matti Turtinen and Juha Vaihoja for the many discussions that contributed to this work.

VIII. References

[1] M. Ackerman and L. Palen, "The Zephyr help instance: promoting ongoing activity in a CSCW system," *Proc. of the CHI '96*, pp. 268-275, 1996.

[2] B. Adelson and E. Soloway, "The role of domain experience in software design," *IEEE Trans. on Software Engineering*, vol. 11, pp. 1351-1360, 1985.

[3] Y. Akao, "An introduction to quality function deployment, " in Y. Akao (ed.), *Quality Function Deployment. Integrating Customer Requirements into Product Design*, Productivity Press, pp. 3-24, 1990.

[4] L.J. Bannon and K. Schmidt, "CSCW: four characters in search of context," in J. M. Bowers and S. D. Benford (eds.), *Studies in Computer Supported Collaborative Work*, Amsterdam: Elsevier Science Publishers, North-Holland, 1991.

[5] G. Barczak and D. Wilemon, "Communications patterns of new product development team leaders," *IEEE Trans. on Engineering Management*, vol. 38, no. 2, pp. 101-109, 1991.

[6] V.R. Basili, G. Caldiera and H.D. Rombach, "Goal Question Metric Paradigm," in J. J. Marciniak (ed.), *Encyclopedia of Software Engineering 1*, New York: John Wiley & Sons, pp. 528-532, 1994.

[7] S. L. Brown and K.M. Eisenhardt, "Product development: past research, present findings, and future directions," *Academy of Management Review*, vol. 20, no. 2, pp. 343-378, 1995.

[8] G. Burchill and C.H. Fine, "Time versus market orientation in product concept development: empirically-based theory generation," *Management Science*, vol. 43, no. 4, pp. 465-478, 1997.

[9] C. Ciborra, "The platform organization: recombining strategies, structures, and surprises," *Organization Science*, vol. 7, no. 2, pp. 103-118, 1996.

[10] R.G. Cooper and E.J. Kleinschmidt, "What separates winners from losers?" *Journal of Product Innovation Management*, vol. 4, no. 3, pp. 169-184, 1987.

[11] R.G. Cooper and E.J. Kleinschmidt, "Benchmarking the Firm's Critical Success Factors in New Product Development," *Journal of Product Innovation Management*, vol. 12, no. 5, 1995.

[12] B. Curtis, H. Krasner and N. Iscoe, "A field study of the software design process for large scale systems," *Communications of the ACM*, vol. 31, no. 11, pp. 1268-1287, 1988.

[13] M.K. Daskalantonakis, "A practical view of software measurement and implementation experiences within Motorola," *IEEE Trans. on Software Engineering*, vol. 18, no. 11, pp. 998-1010, 1992.

[14] O. El Sawy, *Minding Your Own Business Processes: The BPR Learning Book*. New York: McGraw-Hill, 1998.

[15] O.A. El Sawy, G. Bowles G. and B. Kirkwood, "Redesigning the customer support process for the electronic economy: insights from Storage Dimensions," *MIS Quarterly*, vol. 21, no. 4, pp. 457-483, 1997.

[16] I. Foster, M.E. Papka and R. Stevens, "Tools for distributed collaborative environments: a research agenda," in *Proc. of the 5th IEEE International Symposium on High Performance Distributed Computing*, pp. 23-28, 1996

[17] I. Gorton, Hawryszkiewycz. and L. Fung, "Enabling software shift work with groupware: a case study," *Proc. of the 29th Annual Hawaii International Conference on System Sciences*, Maui, Hawaii, 1996.

[18] A.J. Griffin and J.R. Hauser, "Patterns of communication among marketing, engineering and manufacturing, - a comparison between two new product teams," *Management Science*, vol. 38, no. 3, pp. 360-372, 1992.

[19] A.P. Hameri and J. Nihtilä, "Distributed new product development project based on internet and World-Wide Web: a case study," *Journal of Product Innovation Management* vol. 14, no. 2, pp. 77-87, 1997.

[20] J.R. Hauser and D. Clausing, "The house of quality," *Harvard Business Review*, vol. 66, no. 3, pp. 63-73, 1988.

[21] D. Herlea and S. Greenberg. "Using a groupware space for distributed requirements engineering," *Proc. of the Seventh IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 57-62, June 1998.

[22] J.A.Jr. Hrones, B.C. Jedrey and D. Zaaf, "Defining global requirements with distributed QFD," *Digital Technical Journal*, vol. 5, no. 4, pp. 36-46, 1992.

[23] G.P. Huber and R.R. McDaniel, "The decision-making paradigm of organizational design," *Management Science*, vol. 32, no. 5, pp. 572-589, 1986.

[24] A.F. Hutchings and S.R. Knox, "Creating products that customers' demand," *Communications of the ACM*, vol. 38, no. 5, pp. 72-80, 1995.

[25] R. Katz and M. Tushman, "Communication patterns, project performance, and task characteristics - an empirical evaluation and integration in an R&D setting, " *Organizational Behavior and Human Performance*, vol. 23, pp. 139-162, 1979.

[26] D. Leonard-Barton, *Wellsprings of Knowledge: Building and Sustaining the Sources of Innovation*. Boston, MA: Harvard Business School Press, 1995.

[27] P. Lloyd and R. Whitehead, *Transforming Organizations Through Groupware: Lotus Notes in Action*. New York: Springer, 1996.

[28] M. Mandviwalla and L. Olfman, "What do groups need? A proposed set of generic groupware requirements," *ACM Trans. on Computer-Human Interaction*, vol. 1, no. 3, pp. 245-268, 1994.

[29] M.E. McGrath, M.T. Anhony and A.R. Shapiro, *Product Development: Success Through Product and Cycle-Time Excellence*. London: Butterworth-Heinemann, 1992.

[30] I. Nonaka, "A dynamic theory of organizational knowledge creation," *Organization Science*, vol. 5,

no. 1, pp. 14-37, 1994.

[31] I. Nonaka and H. Takeuchi, *The Knowledge-Creating Company*. New York: Oxford University Press, 1995.

[32] I. Nonaka, K. Unemoto and D. Senno, "From information processing to knowledge creation: a paradigm shift in business management," *Technology in Society*, vol. 18, no. 2, pp. 203-218, 1996.

[33] W. Orlikowski, "Learning from Notes: organizational issues in groupware implementation," *Proc. of the CSCW '92: Conference on Computer-Supported Cooperative Work*, Toronto, Canada, pp. 362-369, 1992.

[34] W. Orlikowski, "Improvising organizational transformation over time: a situated change perspective," *Information Systems Research*, vol. 7, no. 1, pp. 63-92, 1996.

[35] G. Osguthorpe, B. Steele and A. Houldcroft, "A method for developing CSCW systems," *Proc. of the 5th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 214-219, June 1996.

[36] D.E. Perry, N.A. Staudenmayer and L.G. Votta, "People, organizations and process improvement," *IEEE Software*, vol. 11, no. 4, pp. 36-45, 1994.

[37] J.B. Quinn, P. Anderson and S. Finkelstein, "Managing professional intelligent: making most of the best," *Harvard Business Review*, pp. 71-80, March 1996.

[38] A. Shepetuk, "Is your product development process a tortoise or a hare?" *Management Review*, vol. 80, no. 3, pp. 25-27, 1991.

[39] P.G. Smith and D.G. Reinertsen, "Shortening the product development cycle," *Research Technology Management*, vol. 35, no. 3, pp. 45-49, 1992.

[40] E.W. Stein and V. Zwass, "Actualizing organizational memory with information systems," *Information Systems Research*, vol. 6, no. 2, pp. 85-117, 1995.

[41] R. Stevens, "Structured requirements," in *Proc. of the Fourth International Symposium on Systems Engineering*, pp. 99-104, 1994.

[42] M.F. Theofanos and S.L. Pfleeger, "Wavefront: a goal-driven requirements process model," *Information and Software Technology*, vol. 38, pp. 507-519, 1996.

[43] J.P. Walsh and G.R. Ungson, "Organizational memory," *Academy of Management Review*, vol. 16, no. 1, pp. 57-91, 1991.

[44] A.H. Van De Ven, "Central Problems in the Management of Innovation," *Management Science*, vol. 32, pp. 590-607, 1986 .

[45] R. Watkins and M. Neal, "Why and how of requirements tracing," *IEEE Software*, vol. 11, no. 4, pp. 104-106, 1994.

[46] A.C. Yeh, "REQUirements Engineering Support Technique (REQUEST): A market driven requirements management process," *Proc. of the Second Symposium on Assessment of Quality Software Development Tools*, pp. 211-223, 1992.

[47] A. Lohva, "*Success factors of the requirements management process and the areas for groupware support*," M.Sc. thesis, University of Jyväskylä, Department of Computer Science and Information Systems, Finland, 1999.

[48] T. Käkölä, "Increasing the interpretive flexibility of information systems through Embedded Application Systems," *Accounting, Management & Information Technologies*, vol. 5, no. 1, pp. 79-102, 1995.

[49] T. Käkölä, "*Dual Information Systems in Hyperknowledge Organizations*," Turku Centre for Computer Science (TUCS) Dissertations, No. 2, University of Turku, Finland, 1996. Also available at http://www.cs.jyu.fi/~timokk.

[50] T. Käkölä and K. Koota, "Dual Information Systems: Supporting organizational working and learning by making organizational memory transparent," *Journal of Organizational Computing and Electronic Commerce*, vol. 9, no. 2&3, pp. 205-232, 1999.

[51] T. Käkölä and K. Koota, "Redesigning computer-supported work processes with Dual Information Systems: the work-process benchmarking service," *Journal of Management Information Systems*, vol. 16, no. 1, pp. 87-119, 1999.

[52] J. Walls, G. Widmeyer, and O. El Sawy, "Building an information systems design theory for vigilant EIS," *Information Systems Research*, vol. 3, no. 1, pp. 36-59, 1992.

[53] Software Engineering Institute, Carnegie Mellon, *"CMMI-SE/SW, V1.0 Capability Maturity Model-Integrated for Systems Engineering/Software Engineering, Version 1.0*," CMU/SEI-2000-TR-018, Pittsburgh, PA 15213-3890, 2000.

[54] P. Trott., *"Innovation Management & New Product Development*," Pearson education limited, Essex, 1998.

[55] R. G. Cooper, "Winning at New Products: Accelerating the Process from Idea to Launch," Third Edition, Perseus Publishing, Cambridge, MA, 2001.

[56] A. Cockburn, *"Agile Software Development,"* Addison-Wesley, USA, 2002.

Tables and Figures

Table 1. Framework for Analyzing Meta-Requirements for RMS

RMS support / RMS use	1. Communication	2. Control
A. Context	• Development and application of domain models to support information storage and retrieval	• Use of domain models in the allocation of responsibilities for the further processing of requirement information
B. Process	• Enforcement of jointly approved milestones for workflow support	• Separation of processes for knowledge acquisition and decision making • Clarification of decision criteria and rules for applying the criteria
C. Content	• Adoption of standard representational schemes in the description of requirements information	• Accumulation of a full revision history with each requirement

Table 2. Phases in Requirements Management.

Phase	Description
Capture	The process of collecting new product ideas and requirements from all relevant sources and with minimal control on representation or content.
Categorization	The association of submitted requirements with appropriate context.
Refinement	The transformation of requirements information into units about which a tentative product decision can be made.
Assessment	The evaluation of requirements information by all relevant stakeholders in view of their alignment with the criteria and targets that have been placed.
Follow-up	Systematic monitoring of requirements usage during the later phases of the product development process.

Table 3. Two-Layered Document Structure.

Document	**Definition**
Product idea	An expression of a customer need or some other desired property that may call for the implementation of a new or enhanced product functionality.
Feature proposal	A description of product functionality expressed in a sufficiently detailed and unambiguous format to permit the making of a go/no-go decision.

Table 4. The Generic Structure of a Requirement.

Class	**Question**	**Attributes**
Description	What is the requirement about?	❍ Description ❍ Rationale
Origin	Where does the requirement come from?	❍ Author ❍ Source ❍ Date of creation
Categorization	What parts of the product and the development organization is the requirement related to?	❍ Traceability links ❍ Position in product structure (c.f., architecture) and associated organizational responsibilities ❍ Interfaces to other product lines
Analysis	What are the implications of the requirement?	❍ Status ❍ Priority ❍ Customer need ❍ Required work effort ❍ Risks
Workflow	What should be done to this requirement next? By whom?	❍ Task description ❍ Assignments to persons
History	What has been done to the requirement? When?	❍ Information about all prior edits, editors, and changes

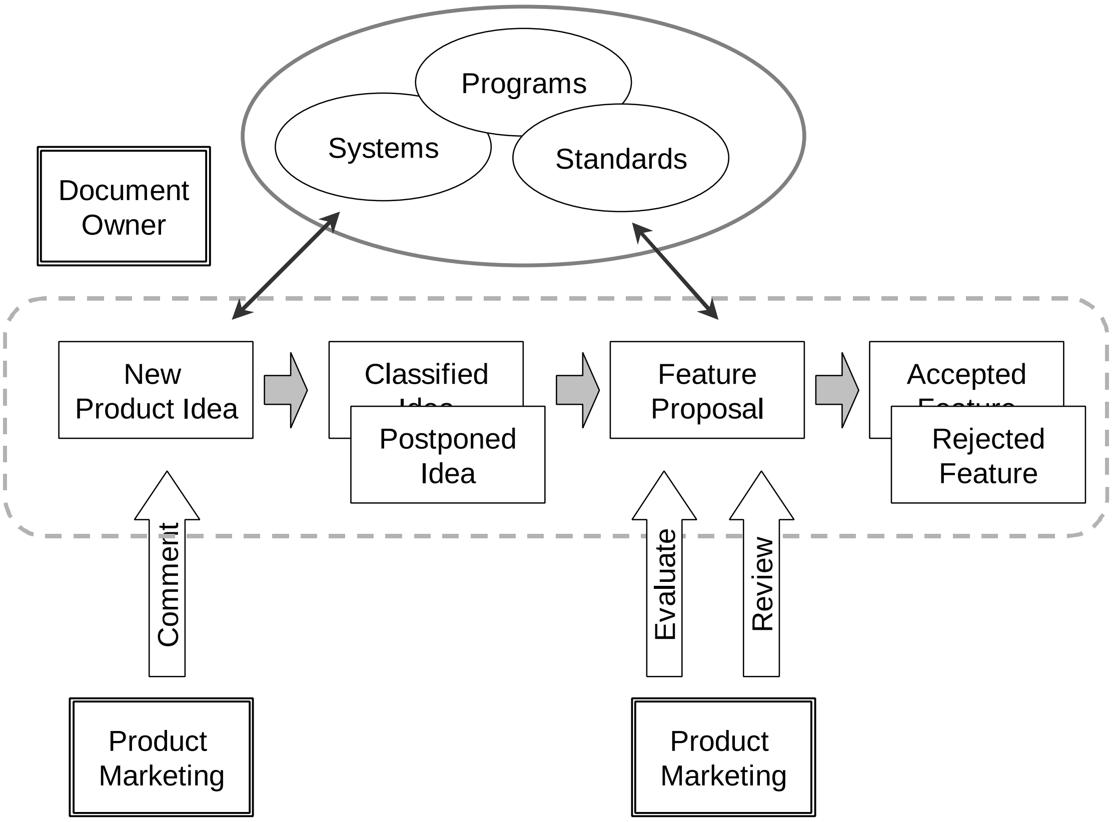
Table 5. Role-, Content-, and Process-Oriented Views for Information Navigation.

Axis of orientation	**Basis of information organization (Examples in italics)**
Role	The role in which the user interacts with the requirements database. (e.g., Requirements sorted by their author)
Content	Content-related assessments which have been made about the requirement. (e.g., Requirements sorted by their priority)
Process	Stage in the document life-cycle. (e.g., Requirements sorted by their review status)

Table 6. Examples of Metrics.

Type	Question
Inputs	❍ How many new product ideas are being submitted monthly? ❍ What percentage of submitted requirements come from customers? ❍ Which functions in the NPD organization create requirements?
Outputs	❍ How many new feature proposals are produced and approved monthly?
Involvement	❍ Which persons and organizations are the most active ones in requirement processing? ❍ How many weekly read and write operations are there? ❍ Who are the key persons in the requirements process?
Efficiency	❍ How long does it take - on the average - to develop and approve specifications for a new product functionality? ❍ What percentage of suggested functionalities is approved (i.e., what is the decision yield)?

Figure 1. A Process Illustration.



AHTI A. SALO is a professor at the Systems Analysis Laboratory, Helsinki University of Technology where he obtained his D.Tech. degree in systems and operations research. He has worked as a research fellow at the London Business School and as a senior consultant at the Nokia Research Center. His research interests include the application of decision analytic tools and information systems in product development and technology management. He has published extensively on these topics, with articles in leading journals such as *International Journal of Technology Management, Technological Forecasting and Social Change* and *Journal of Multi-Criteria Decision Analysis*.

TIMO K. KÄKÖLÄ is a professor of information systems and software business at the Department of Computer Science and Information Systems, University of Jyväskylä, Finland. He has a Ph.D., Ph. Lic., and M.Sc. degrees in computer science and information systems from University of Turku and an M.B.A. in management from Helsinki School of Economics and Business Administration, Finland. His research interests include IT-enabled organizational designs for effective organizational and interorganizational creation and sharing of knowledge, the design of information systems architectures that support the learning, enactment, breakdown management, and redesign of knowledge-intensive work processes, and the business models and processes of software-intensive product-oriented ventures. He has published a number of articles in leading scientific journals and conferences.

Groupware Support for Requirements Management in New Product Development

Index terms

Computer-mediated communication

New product development

Decision support systems

Groupware

Information systems design theory

Knowledge management

Organizational learning

Process maturity

Requirements management

Shared knowledge creation.

Page: 10

[TKK1] Numerous social theories can serve as kernel theories of RMS but no single theory will ever be adequate. Theories should be general and accurate to be useful but they should also be simple to have broad applicability in practise. Thorngate's [54] postulate of commensurate complexity states that a social theory cannot be simultaneously general, accurate and simple. Only two of the three characteristics can be realized at any given time [54]. A set of interconnected theories can be devised so various stakeholders involved in RMS design can leverage them effectively. For example, general and simple theories could be used to *guide* those aspects of RMS design where the views of all stakeholders are necessary and need to be integrated. However, according to the postulate such theories are necessarily abstract and not accurate, thus having limited presciptive power over RMS meta-design. General and accurate theories In addition to these three common requirements for kernel theories, there are more RMS specific requirements.

Page: 10

[TKK2] Because of the dual nature of technologies, information systems research must put equal emphasis on design and *use* of information systems: information systems design and use theories are needed. I have proposed kernel theories of DIS such as structuration theory that span design and use processes, and thus transcend the design focus of design theories.

Information systems researchers must deal with a new level of theoretical and empirical complexity to properly account for information systems use in theory development. DIS encourage both theoretical reflection and active improvisation in the enactment, breakdown management, and improvement of routines. The extent to which *prescriptive* theories can be extended to govern the use of DIS is thus an interesting problem for future research.