JITTA
JOURNAL OF INFORMATION TECHNOLOGY THEORY AND APPLICATION

COMMUNICATION FLOWS IN SOFTWARE PRODUCT DEVELOPMENT: A CASE STUDY OF TWO MOBILE SOFTWARE FIRMS

TUURE TUUNANEN, The University of Auckland
The University of Auckland Business School, The Dept. of ISOM, Symonds Street 7, Private Bag 92019, Auckland, New Zealand, Tel: +64-9-373-7599 ext. 84622, Fax: +64-9-373-7430, Email: tuure@tuunanen.fi

MARIANNE VAINIO, Helsinki School of Economics
Department of Business Technology, P. O. Box 1210, FIN-00101 Helsinki, Finland, Tel: +358-50-529-4723, Fax +358 9 431 38700, Email: marianne.vainio@hse.fi

ABSTRACT

After a steady rise in the revenue that they achieved in the late 1990s, small software firms, in particular, were hit hard. Because of the restricted investments and tight budgets, the goals of firm strategies started shifting towards delivering outstanding products in terms of price, quality and performance. However, the focus of current information systems (IS) development methods is on improving processes to produce better and more predictable results, and, therefore, they tend to lack responsiveness to market opportunities. In this study, we review the marketing-related discussion of new product development (NPD), and suggest that the NPD framework offers valuable insights for the development of mobile software products. The NPD framework especially contributes to interactions with other business dimensions and the firm's environment. In an interpretive case study of two mobile software firms, we apply both the NPD approach and the IS development methods as a lenses to identify the participants involved in the development of software products, and how the information was communicated between them throughout the phases of software product development. In the two firms, the applied framework uncovered the communication flows between the participants of software product development and integrated the interaction between them into a coherent view. In particular, the findings indicate the importance of informal, tacit communication as a basis for these interactions. As a result of this study, a preliminary conceptual model is presented describing the integration of the NPD approach with the IS development methods through cross-functional teams and rich communication in the development of software products. As a contribution, we suggest that the

Ken Peffers acted as the senior editor for this paper.

Tuunanen, T. and M. Vainio, "Communication Flows in Software Product Development: A Case Study of Two Mobile Software Firms," *Journal of Information Technology Theory and Application (JITTA), 7*:3, 2005, 27-48.

integration of the marketing-related NPD framework with the IS development methods provides guidance for managers to develop successful mobile software products in the dynamic markets in which small software firms exist.

INTRODUCTION

Mobile devices are increasingly becoming standardized commodity products instead of exclusive high-technology products due to the falling retail and service prices. Thus, the firms operating in the field have started to look more closely at their software product development and to look for means that would yield more new outstanding quality products for less investment. This shift also stresses the need for effectively integrating the knowledge from various sources, such as the market, customers, users, competitors and regulatory parties, in order to enhance the market responsiveness of their products. We present that there is a need for integrating more information from the markets to the development of mobile software products.

The information systems (IS) literature has been viewing the relationships with customers and users as transaction-oriented and based on specific organizational projects. Several of the early approaches to IS development methods, like the Waterfall (Royce 1970), suggest that the user needs be collected at the beginning by the analysts. Although researchers have suggested some more iterative methods of development (e.g. Boehm 1988), the linear way of thinking is still quite dominant in the current ways of collecting software requirements (Mathiassen, Saarinen, Tuunanen and Rossi 2004). Among IS development methods, incorporating market elements into the process has proven especially difficult (Regnell, Hösta, Dag, Beremark and Hjelm 2001), as the current views tend to overlook up-front business planning (Vainio, Tuunanen

CONTRIBUTION

This paper makes a contribution to the information systems (IS) research of development methods. The focus of current IS development methods is on improving processes to produce better and more predictable results, and, therefore, they tend to lack responsiveness to market opportunities. The marketing-related discussion of new product development (NPD) provides valuable insights for the development of mobile software products as it especially contributes to interactions with other business dimensions and the firm's environment. This study is the first one to attempt to introduce market elements into the design of development of software products. Further, we consider the NPD concepts and processes as a way of bridging the gap between what is currently done in software development and what we believe needs to be done. The empirical evidence of this study demonstrates how the different participants were involved in the development, and how they communicate throughout the phases of software product development. In both studied firms, the integration of the NPD with the IS development methods helped us integrate important communication flows of software product development into a coherent view. In particular, the study points out the importance of informal, tacit communication between the participants as a basis for these relationships. As a synthesis of our findings, we present the conceptual model of software product development. This model contributes to gaining better understanding of how different participants relate to the market success of a software product and helps manage these relationships.

This study is expected to be very interesting both to researchers and practitioners. For researchers our study opens up the problematic of the integration of marketing with software development. Furthermore, our study can be of interest to IS researchers focusing on agile development methods as we approach organizational agility from the marketing-related perspective yet staying in the software field. Finally, the studies findings provide guidance for practitioners to assess and develop more market-oriented software products in contemporary dynamic environments.

and Abrahamsson 2004).

Within marketing science, the problem of developing innovative new products has led to the birth of a new specific discipline, called "new product development" (NPD), which focuses on delivering a product from idea to launch. The NPD literature presents that customers and users are important sources of innovation and firms are therefore encouraged to commit considerable resources to build sustainable, long-term relationships to them. Researchers have especially stressed the importance of listening and responding systematically to the voice of the customer (Cooper 2000; Hauser 1988; Zahay, Griffin and Fredericks 2004). Furthermore, firms need to understand their external environment and integrate this knowledge appropriately with their knowledge domain. Functional integration of marketing and design functions has been said to be one of the key issues for the success of products (Barczak 1995; Gupta 1985; Souder 1988). Similar to the IS research, few researchers have emphasized a well-structured process (Cooper 1990), arguing that by executing an NPD with incremental commitments, firms can minimize the market risk involved.

Basically, software product development can be said to be a knowledge-activity (McGrath 1996). We present that institutionalizing and leveraging knowledge and experience of stakeholders will increase the responsiveness to market opportunities. Communication between stakeholders is likely to be even more important in order to produce market-oriented software products. This need of communication has been widely recognized within the IS field (Keil and Carmel 1995; Tuunanen 2003), but no exact ways of facilitating the information flows during software product development have been proposed. On the other hand, fostering a culture of communication and functional integration provides more flexibility to respond to the dynamic environment (Mata, Fuerst and Barney 1995). In this study, we review the marketing-related discussion of new product development (NPD), and suggest that the NPD framework offers valuable insights for the development of mobile software products. The NPD framework especially contributes to interactions with other business dimensions and the firm's environment.

We used an interpretive case study of two mobile software firms (Klein and Myers 1999; Orlikowski and Baroudi 1991) as a research methodology to demonstrate the integration of the marketing-related NPD development with the IS development methods. By using both the NPD approach and the IS development methods as a lenses we identified the participants involved in the development and how the information is communicated between them during the phases of software product development. The findings address the importance of continuous communication across software product development. We also show that while formal communication proved to travel according to the pre-defined process specifications, the essential information for product development was exchanged in a more informal way. In particular, we identified three main information flows in the development of software products. At those points, teams got meaningful, high-fidelity feedback on the performance of the product and undertook responding to that information. As a synthesis of our findings, a preliminary conceptual model describing these identified communication flows is presented.

The structure of the paper is as follows. Next, we review insights from the IS literature to identify the development methods suitable for mobile products. Then, we review how the NPD discipline has addressed similar issues. As a summary, an assessment of the two approaches is presented to guide our empirical research. This is followed by the design of the interpretive case study. Subsequently, we present two cross-case analyses with a synthesis of the findings. Finally, we discuss the results and present the conclusions.

IS DEVELOPMENT METHODS

In the IS literature, software development methods have evolved from disorganized ways of working towards more organized ones. The 'waterfall model' was one of first ones to emerge to the scene (Royce 1970). The waterfall model is still well-used and it can be described as a systematic, sequential approach, in which each stage requires well-defined input, and results in well-defined outcomes.

The software product is not delivered until the whole linear sequence has been completed. Royce (1970) suggests repeating this linear sequence at least twice. The problems related to this linear model are stagnant requirements and badly structured programming. In an attempt to avoid the problem, overlapping was implemented between the stages (Boehm 1988). However, the linear method has been argued to be too mechanistic (Nandhakumar and Avison 1999) and merely idealistic providing only normative guidance for development situations (Truex, Baskerville and Travis 2000).

In contrast to the waterfall model, an evolutionary approach to software development is often more effective in producing systems that meet the immediate needs of customers. The specifications are developed incrementally, while reflecting the user's understanding of software problems (Boehm 1988). In the evolutionary approach, reuse is often seen as essential for rapid software development. Furthermore, there is a need to support process iteration where parts of the process are repeated as system requirements evolve. Incremental development and spiral development models meet this requirement. In incremental development, software is developed in small but usable units, which can be separately delivered to the customer. Each increment is an operative subset of the system and builds on the increments that have already been made (Pressman 2000). Detailed design, coding, and testing occur within these separate stages (McConnell 1986). The process of development, validation and integration continues until the delivered increments form a complete product.

The rational unified process (RUP) is a contemporary example of iterative and incremental methods. This popular model has been argued to take different aspects varying from traditional plan-driven approaches to new agile ones, e.g. (Merisalo-Rantanen, Tuunanen and Rossi 2005), depending on project characteristics. Unlike many of the traditional software process descriptions, RUP places high emphasis on the business context of the project. Whether the software is produced for a given customer, to be put on the market or to be developed for an internal customer, the business modeling done during the inception and elaboration phases can be adjusted according to the purpose for which the software is built (Kruchten 1996). The RUP process can be approached from two different and integrated perspectives: 1) a management perspective, dealing with financial, strategic, commercial, and human aspects; and 2) a technical perspective, dealing with quality, engineering and design method aspects.

Agile development methods are another approach to the problems concerning understanding users' needs (Abrahamsson 2003; Abrahamsson, Warsta, Siponen and Ronkainen 2003). Extreme programming (XP) is an example of the agile software development methods that have emerged in the past few years. XP was first introduced in (Beck 1999a). Agile methods challenge the traditional models alleged to be too mechanistic, and approaches software development with such values as: individuals and human interactions over processes and tools, working software over comprehensive documentation, intense customer collaboration over contract negotiation, and responding to change over following a plan (Beck, et al. 2001). According to Highsmith and Cockburn (2001), agile methods recognize people as the primary drivers of project success, coupled with an intense focus on effectiveness and maneuverability. The core of agile software development methods is defined as the use of light-but-sufficient rules of project behavior and the use of human- and communication-oriented rules (Cockburn 2002). Researchers have presented that XP is a combination of best practices of more traditional software development methods (Merisalo-Rantanen, Tuunanen and Rossi 2005).

NEW PRODUCT DEVELOPMENT

New product development is generally defined as a set of activities that transform new product ideas into new product designs. The NPD system encompasses the NPD process, the management and support of this process; the technologies incorporated in people and resources or means needed to carry out these processes; and the organizational arrangements used to divide and co-ordinate the processes. It is an open system, interacting with its internal and external environment through its inputs,

outputs and resources (Chiesa, Coughlan and Voss 1996). The adoption of NPD is said to speed up time to market, improve product quality as well as increase development efficiency, build core competence, and increase innovative ability.

Like in the IS literature, NPD researchers have found that a complete, formal product development process potentially enhances the utilization of information and the effectiveness of decision-making (Clark and Wheelwright 1993; De Maio, Verganti and Corso 1994; Hart and Baker 1994). Cooper (1983; 2000), has suggested that this process should involve a wide variety of tasks and activities, rather than be reduced to a few dominant stages. Additionally, such key activities as screening, market research, customer trials, and market launch should not be forgotten in the process. There should also be a balance between market oriented and technical activities. Cooper has elaborated these ideas by presenting a particular development method: the StageGate model (Cooper 2000).

Figure 1 illustrates the StageGate model. The process description has been simplified to five stages representing the major events in the NPD process (Cooper 2000). The stages are cross-functional, thus there is no R&D or marketing stage. Each stage consists of a set of parallel activities undertaken by people from different functional areas in the firm. The players of the project team undertake key tasks to gather information needed to advance the project to the next gate or decision point. The gates between the different stages serve as quality control checkpoints. To manage risks via the StageGate method, the activities in a given stage must be designed to gather vital information – technical, financial, and operation-specific – in order to drive down the technical and business risk. The plan is based on incremental commitments, as each stage costs more than the previous one. The gates also have a common format, which includes deliverables, criteria and outputs.

Another interesting perspective to the NPD methods is the Generic Development Process by Ulrich and Eppinger (2000). According to them, the typical phases of generic product development are: Planning – Concept Development – System-Level Design – Detailed Design – Testing and Refinement – and Production Ramp-up. The authors illustrate the development process as an information-processing system. The process begins with inputs such as corporative objectives and the capabilities of available technologies, product platforms, and production systems. Various activities process the development information, formulating specifications, concepts, and design details. The process concludes when all the information required to support production and sales has been created and communicated. The generic development process model (Ulrich and Eppinger 2000) identifies the different functions of an organization, such as marketing, design, and manufacturing, during each development phase. The model also calls for tremendous integration across the functions of the development team. The front-end process of concept development, in particular, requires more coordination among the functions than any other phase.

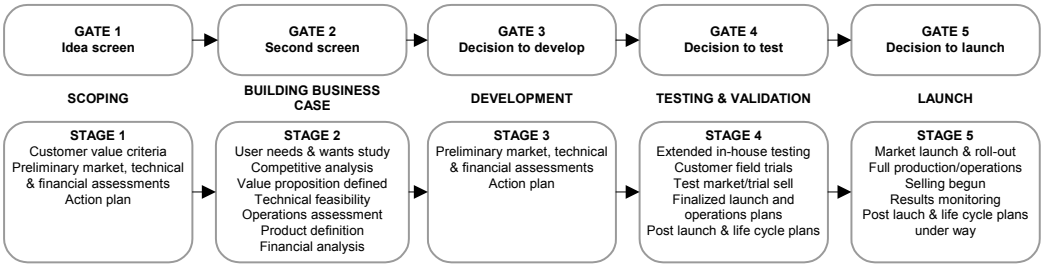

Figure 1. The Stage Gate model, modified from Cooper (2000)

ASSESSMENT FOR SOFTWARE PRODUCT DEVELOPMENT

Although the IS development methods and the NPD ones excel in their own domain areas, there is no a comprehensive view to software product development. In the IS development, the process management focuses on identifying risks and drawing up plans to minimize the effect of risks on the project (Alter and Ginzberg 1978; Davis 1982; McFarlan 1981). The focus is on developing individual products faster with a better process producing more quality and more predictable results. Although the literature specifies several causes for uncertainty in software development (Barki, Rivard and Talbot 1993; Boehm, Clark and al. 1995), the main identified risk categories are project-related, affecting project schedule and resources, and technological, affecting the quality or performance of the software application being developed. From the IS perspective, business risks affect the organization developing the software, which indicates that business risk management is not incorporated as such into the existing software process models. To summarize, it can be suggested that the IS process approach ignores the emphasis on up-front business planning and late-phase product launch preparation, making the software product development unbalanced (Vainio, Tuunanen and Abrahamsson 2004).

In response to the lack of market responsiveness, the NPD literature has provided development methods that attempt to build success into the process by designing stages for gathering the market information needed to lower the business risk (Cooper 2000). Each stage costs more than the previous one, the model is thus being based on incremental commitments. The purpose is to move products from concept to market faster and more efficiently. However, the popular stage-gate approach used in many software companies is rather similar to the linear waterfall process, which can make the accommodation of the newest agile methodologies difficult. This suggests that software firms should require more than just a series of stages. It might be useful to adopt ideas from Boehm (1988), for example, and study if product development processes could be extended to allow for iteration and experimentation.

In software product development, information is clearly a resource that is necessary for development teams. How this information is managed in a firm is important (Zahay, Griffin and Fredericks 2004) and can produce a competitive benefit. Since software development is basically a knowledge-activity (McGrath 1996), we can argue that institutionalizing and leveraging knowledge and experience increases productivity. By encouraging these activities, communication is likely to lead to a greater degree of integration of the various functions in an organization. As (Mata, Fuerst and Barney 1995) summarize, successful companies increase flexibility in their organizations by fostering a culture of communication and functional integration. This need of communication has also been widely recognized within the IS field (Curtis, Kellner and Over 1992; Keil and Carmel 1995), while no exact ways of facilitating the information exchange between marketing and software development have been proposed. We therefore take the reviewed literature as our research lenses and use it to demonstrate and interpret how 1) different participants are involved in the development, and 2) how the information is communicated during the phases of software product development.

RESEARCH METHODOLOGY

This research is designed as a deductive, interpretive case study to explore the development of mobile software products in high velocity markets. Specifically, we apply the NPD development and the IS development methods as a research lenses to identify how the different participants were involved in development of software products, and how the information was communicated between them throughout the phases of software product development. To focus the study more accurately and to help shape the design of the study (Eisenhardt 1989; Klein and Myers 1999), we use the chosen research lenses to create a firmer empirical grounding for drawing implications. We adopt an interpretive approach (Walsham 1993) to study how the participants of the development of software products interact in small mobile software firms. Interpretive studies attempt to

understand phenomena through accessing the meanings that participants assign to them and to explain why members of a social group act the way they do (Klein and Myers 1999; Orlikowski and Baroudi 1991). Therefore, the chosen research methodology is in line with our research objectives.

Case Selection

We use the case study approach (Yin 1994) to understand the dynamics in the development of mobile software products. The approach includes different data collection methods and provides many sources of evidence to help reach a deep understanding of the phenomenon and to ensure satisfactory validity of the findings (Eisenhardt 1989; Yin 1994) . Similar to other studies (Barley 1986; Heaton 1998; Orlikowski 1993), our case selection was based on the theoretical sampling to obtain information from comparable cases (Glaser and Strauss 1967; Orlikowski 1993). Replication adds confidence and robustness to the findings, but it does not ensure generalizable results (Miles and Huberman 1994; Yin 1994). We selected two small, Finnish software firms that both seek high growth based on initial venture capitalists' investments. The two firms operate in similar contexts and have similar goals, but they are in different stages of business development. Thus, the firms were selected for their similarities as well as their differences.

The identified two firms are competent in responding to the market opportunities with their mobile software products. From 1999 to 2003, their turnover has increased steadily. Also, the number of customer accounts, the size of the organization, and investor funding increased yearly in both firms. These indicators suggest that both firms were capable of developing successful software products. The selection strategy used thus ensured that the firms were being worthy of examination in this study. Equally, we wanted two firms on different stages of business development. The literature suggests that the maturity of a firm affects its development practices (Dove 2001; Greiner 1998). One firm, *Multimedia,* is a recent start-up, and the other, *Messaging*, is a mature, but still small software firm. As a result, the two firms differ on dimensions such as size, available resources, management and culture. These differences in organizational conditions allowed us to make contrasting interpretations during data analysis.

Data Collection

To ensure rich data from the two firms and to facilitate triangulation, we collected evidence from four sources within each firm (Yin 1994). *First*, we acquired written material including brochures, annual reports, internal documents, and trade journal articles about the firms. *Second*, we used archives such as marketing presentations, organizational records, project documentation, and customer records. In doing so, we kept in mind that the documents stated the interpretations of actions in the firms rather than unmitigated truth (Yin 1994). *Third*, we used observation through site visits. One of the authors visited both firms three times and made observations of meetings, locations of work activities, and normal office communications. These observations provided valuable information about organizational arrangements and practices. *Fourth*, we conducted theme interviews of three types in each firm as the use of multiple respondents enhances the creative potential of the study and builds confidence in the findings (Eisenhardt 1989).

To begin with, the CEO (at *Multimedia*) and the head of business development (at *Messaging*) gave us details focusing on management, business development, and marketing issues. Subsequently, these key informants suggested managers involved in such practical operations to join for additional interviewing to offer more details. Finally, the head of software development or the head of technology provided us with details concerning software development practices.

In all interviews, the stream of questions was fluid rather than rigid (Rubin and Rubin 1995) and the interviews were of an open-ended nature. However, to ensure sufficient support for exploring our research theme we focused on a certain set of questions derived from the study protocol (Merton, Fiske and Kendall 1990). The interviews were taped and notes were taken simultaneously. The use of multiple sources of evidence allowed us to address a broader range of historical, attitudinal, and behavioral issues. In particular, the two written sources of evidence were

helpful in corroborating and augmenting evidence from other sources. For example, we compared the development process documentation with interview data to verify the effective involvement of participants in software product development. Thus, we were less likely to be misled by single sources and more likely to be critical in interpreting the contents of each source of evidence.

Selected Cases

The two firms selected operate in the mobile markets and both of them have already a number of references in the markets. The firms differ from one another in terms of business development stage; while one – *Multimedia* – is a start-up firm established in 2001 and it has just recently begun to commercialize the first version of a product, the other – *Messaging* – has already launched four product versions during its six financial years. In 2003, the turnover of *Multimedia* was 2.5 million euros and it had 62 employees. The comparable figures for *Messaging* were 5 million euros and 65 employees. In 2003, *Multimedia* was just about to reach breakeven and was estimating its turnover to be five times as high in 2004. Likewise, *Messaging* was estimating its turnover to grow approx. 20 to 30% in the following year. The selected cases were named *Multimedia* and *Messaging* after their product offerings.

Messaging had developed a middleware solution enabling mobile operators to increase their control in the mobile content business by managing the process of provisioning, delivering, and charging for a service portfolio. Multimedia Framework, as the product of *Multimedia* was called, was a complete system incorporating all the technology required for implementing mobile video applications for mobile devices ranging from cellular phones to digital cameras. The customer segments of the two firms consisted of component suppliers, device and phone manufacturers, operators, and service providers.

Data Analysis

The purpose of our data analysis was to identify and evaluate communication flows throughout the development of software products with a particular emphasis on interactions between the participants involved. In a first rough analysis, we studied financial information from 2000 to 2003, future estimates for 2004, strategy and operating plans, organization structure, and product white papers to focus and plan the detailed data collection through interviews and site visits. Then, in line with (Eisenhardt 1989; Miles and Huberman 1994; Yin 1994), we analyzed these detailed data from different perspectives.

First, we conducted an analysis to identify the participants involved in each development phase and the communication flows between them in each firms. In this study, participants are parties who have an interest in the product, while also having some demands on the product, and who, therefore, are to be consulted in the requirements gathering process. Identifying necessary participants is important because if they do not or cannot accommodate their concerns to the concerns of the product, then the product will likely fail. This cross-case analysis was theory-driven (Miles & Huberman 1994) primarily using the NPD approach and the IS development methods as a lenses to study the interactions of the participants and their nature during the phases of development. The quotations for this analysis are presented directly from our field notes (Lee 1989; Orlikowski 1993), see Appendix 1.

Second, we conducted another cross-case analysis to identify the level of intentional involvement for each participant of the development. Again, this analysis was theory-driven using McGrath's notation (1995) to identify the contribution of participants involved. The CEO of each firm was asked to define the level of involvement of participants. The purpose of the analysis was to describe the involvement of participants by objectives in each phase, and study how it corresponds to other field notes and observations.

Finally, the data collected in each firm was used for a detailed analysis in which we formed a conceptual model reflecting the development phases and the connecting links between the phases in both firms. The NPD approach and the IS development literature were employed as a basis for the conceptualization.

The first and the second analyses help to compare and contrast the interactions and their nature across the two firms. Meanwhile, the conceptualization identifies the similarities regarding the development of software products of the firms. Analyzing data on these distinct perspectives helped ensure a good cross-case comparison. In line with (Eisenhardt 1989; Miles and Huberman 1994; Yin 1994), this systematic approach to a priori specification of model made it possible to develop a coherent understanding of communication flows in the development of software products in the two firms.

FINDINGS

In the following sections, the results of the analyses are presented. First, we present the analysis to identify the participants involved in the development of software products and the communication flows between them. Second, we introduce the other cross-case analysis to identify the level of involvement by objectives in each firm. Finally, we synthesize our findings into a preliminary conceptual model describing the communication flows in software product development.

The decision-making of product innovation in both firms turned out to be coordinated on a milestone basis, which resembled the stage-gate product development process (Cooper 1990). However, both firms had adjusted the high level staged process to accommodate the iterative software development practices. These changes involved incorporating iterations into the middle phases of the process, as the implementation and component factory' phase was modified to emphasize a critical exercise of iteration planning. This phase now broke out into series of planned, structured iterations, which reflected a software product's component-based architecture. As regards the IS methods, *Messaging* had adopted the RUP model whereas *Multimedia* relied on extreme programming. In conclusion, we identified five comparable phases in each firm: 1) scoping and requirement elicitation; 2) building business case and requirement analysis; 3) implementation and component factory; 4) testing and validation / product integration and 5) launch.

Table 1 and 2 provide summaries of our analysis to identify the participants involved in each development phase and the nature of their interactions. In particular, we aim at recognizing: 1) the source of information, while illustrating how the information is identified and from where it is gathered, 2) the nature of the information exchange ranging from informal and tacit to formal and explicit, 3) the activities, while informing how the information is processed and analyzed, and 4) the likely output of a phase in light of information about how the information is distributed during the development cycle.

At *Multimedia*, the decision-making was led by the management group including members from all important functions. *Multimedia* was not employing any systematic approach to forecast end-users' needs and desires. Scoping and requirement elicitation was based on the intuition of the management group and some unofficial discussions with few customers and members of technology forums. Therefore, it can be characterized one-way communication. On the other hand, they also discussed a lot with few of their key customers and business acquaintances. However, the management group rarely produced any systematic and explicit output of their decisions in the first phase. Furthermore, the line between the first phase and the second one was very thin and decision-making during both of the phases relied mainly on the management group's work. However, in the second phase, *Multimedia*'s sales team discussed a lot with their key customers, which provided information for making a decision whether to start development. This material consisted of financial analyses, initial technology feasibility assessment, software development plan, and assessment of the architecture.

Implementation and component factory was performed in-house and there were no external functions involved. The output of the phase consisted of prototypes and early versions of a product. After technical development work, software engineers performed extended in-house testing. From testing and validation to the last phase of launch, feedback was gathered only from few potential customers in casual sales meetings. Early versions of the product were used to simulate a product concept for sales support. These prototypes

served salesmen more than software engineers, as the purpose was to conceptualize the product to collect the customer feedback. It was notable that the product documentation and brochures were published later or not at. The findings also indicate how the nature of information exchange between the participants was based only on informal, face-to-face communication during the last phases. At the launch phase, they contacted potential clients without any formal post-development plans. Nothing but a few pieces of information exchanged or produced were explicit, such as occasionally made system documentations.

Table 1. Communication flows at *Multimedia*.

Stages	Information sources	Two-way/ One-way	Nature of information exchange	Activities by participants
Scoping & Requirement Elicitation	Potential customers, partners, technology forums	One-way	Explicit information (news, documents, info archives), implicit information	Preliminary investigation made by the management group
	Customers, partners	Two-way	Tacit knowledge, casual discussions	Feedback collected by the sales team
	Output: A prelim market assessment in the mind of the management group No decisions			
Building Business Case & Requirement Analysis	Employees	Two-way	Tacit knowledge, casual discussions	Ideas collected by the management group
	Key customers	Two-way	Tacit knowledge, structured discussions	Discussions about the product concept by the sales team
	Output: Financial analyses, technical feasibility assessment (not in explicit form), software development plan, software architecture Decision to start development			
Implementation & Component Factory	-			
	Output: Rapid prototypes, components, evaluation of early output Decision to move to external testing			
Testing and Validation & Product Integration	Potential customers	Two-way	Tacit knowledge, structured discussions, prototype, the first versions of a product	Delivering customer field trials The sales team presents the product to potential customers
	Output: Transition plans, fully functional system No formal decisions – gradual transition to launch			
Launch	Potential customers	Two-way	Tacit knowledge, casual discussions Explicit presentation	The sales team presents the product to potential customers
	Output: Occasionally made system documentation			

Table 2. Communication flows at *Messaging*.

Stages	Information source	Two-way/ One-way	Type of information collected or exchanged	Mechanism of collecting information
Scoping &Requirement Elicitation	Potential customers, partners, technology forums	One-way	Explicit information (news, documents, info archives), implicit information	Preliminary investigation made by the management group
	Current customers, partners	Two-way	Tacit knowledge, casual discussions	The sales team is in contact with customers
	Current customers	One-way	Emails, feedback form	Collecting feedback through a digital channel and storing data in a database.
	Key customers	One-way	Structured survey	A market analysis
	Competitors	One-way	Explicit documents	Systematic collection of bids
	Current customers	One-way	Structured survey	A quality survey
	Technology forums, software seminars	Two-way	Tacit knowledge, casual discussions	The product steering group
	Output: Prelim financial & business analyses, technical assessment, initial domain model analysis, action plans for the next phases Decision to start extensive investigation			
Building Business Case & Requirement Analysis	Financial data, market data	Two-way	Explicit analysis	Detailed investigation made by the management group
	Technology forums, competitors	Two-way	Tacit knowledge, informal discussions	Detailed investigation made by the product steering group
	Key customers	Two-way	Casual discussions, early version of a product concept	The sales team collects feedback from customers
	Output: Financial analysis, value proposition defined, competitive analysis, technical feasibility assessment, domain analysis model, software development plan, software architecture Decision to start development			
Implementation & Component Factory	Key customers	Two-way	Prototype	The sales team collects initial customer feedback by delivering prototypes
	Output: Rapid prototypes, components, evaluation of early output Decision to move to external testing			
Testing and Validation & Product Integration	Key customers	Two-way	Tacit knowledge, structured discussions, prototype, the first versions of a product	Customer field trials. The sales team is in contact with customers and starts to sell
	Potential sales partners	Two-way	Tacit knowledge, structured discussions	The head of business development with marketing positions the product in partner network
	Output: Transition plans and operation plans, test beds and test suites, fully functional system No formal decisions – gradual transition to launch			
Launch	Potential customers, partners and members of technology forums	Two-way	Tacit knowledge, casual discussions Explicit presentations	The management, sales and marketing present the product to potential customers
	Current customers	One-way	Structured survey	After sales survey
	Output: Post-launch & life cycle plans under way, internal feedback, system documentation			

Messaging was clearly more systematic and interactive regarding communication. Similar to *Multimedia*, their management group was responsible for leading common decision-making. However, a product steering group produced additional material for technical, software engineering and product feasibility assessments.

At the scoping and requirement elicitation phase, the most of communication at *Messaging* was colloquial discussion between salesmen and customers. The management group and the product steering group processed this information. They also systematically screened various sources, such as customers, network partners, and competitive environment through competitor offers and market follow-up. Customer feedback was formally collected through a digital channel and stored in a database. The responses were given priorities according to three levels and the results were regularly reviewed by the management group and the product steering group. The quality survey was also conducted for current customers yearly. The output of the phase included prelim financial and business analyses, technical assessment, initial domain model analysis and action plans for the next phase. This information was used for making a decision to start extensive investigation.

In the second phase, *Messaging* continued their extensive communication with customers, partners and competitors to test the feasibility of a product concept. The findings were articulated by explicit means. Already at this phase they started profound co-operation with their key customers with the purpose of gathering in-depth feedback of a product concept. As a product got more completed, they were able to test both the functionality and the usability with a few key customers from the business perspective. The co-operation with customers was characterized as relaxed and friendly communication supporting learning activities of both partners. This phase created diverse material to make the next decision to start development.

Similar to the two first phases, the two-way communication with customers was intense in the phase of implementation and component factory. Their sales team collected initial customer feedback by delivering prototypes. The iterative development according to the RUP guidelines continued until a product was considered to be ready for external testing. At the testing and product integration, *Messaging* started systematically contacting potential sales partners to strengthen their market position. The output of this phase consisted of transition and operation plans, test beds and suites, and fully functional system components. This was followed by gradual transition to launch. Finally, at the launch phase, salesmen and the management presented the product to the market and customers in industry fairs. Subsequently, the results of the launch were monitored and used for future development. As opposed to *Multimedia*, *Messaging* invested a lot in introducing their product to potential partners to be able to form a comprehensive offer with them. They believed that relationships with complementarities will increase their initial market reach and accelerate the sales growth. The relationships were mainly based on personal commitments between the partners as there were no official, explicit agreements.

As a summary, this analysis indicates how the quality of information generated during development differed; *Messaging* collected more intensively richer data from the market. Additionally, the information sources were more plentiful and the direction of communication was more two-way. Although both firms provided the response to the information during development, *Messaging* had defined and structured more explicit and formal activities to support the communication and information management. Testing and validation of a product innovation were more intense at *Messaging* because they had the possibility to gather feedback from more parties.

The second analysis presents how the different participants were involved in the development of software products. The analysis was carried out according to McGrath's notation of concurrent engineering. The results are presented in Figure 2 and 3. The purpose of the analysis was to identify an intentional level of involvement of each participant in each of the development phases. The analysis uses three shades of color to distinguish how different parties contributed to

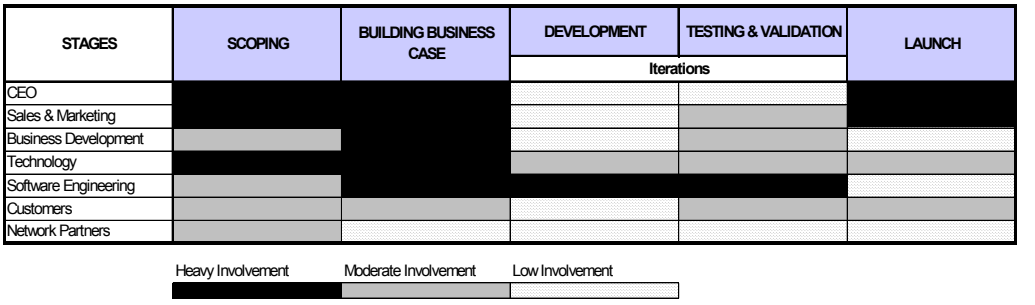

Figure 2. The participants involved in the development of software products at *Multimedia*.

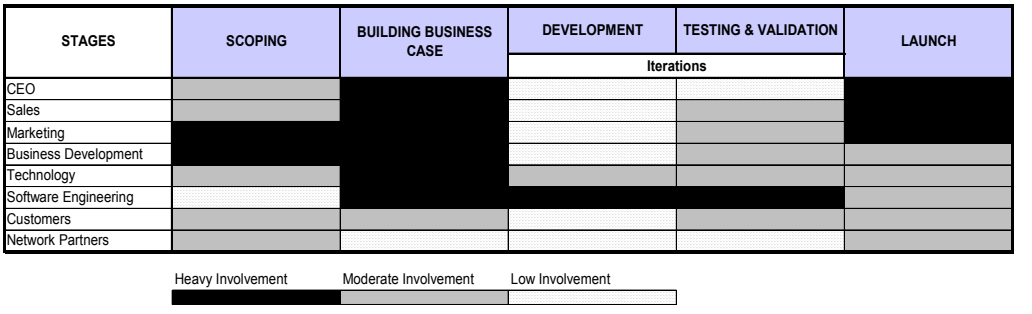

Figure 3. The participants involved in the development of software products at *Messaging*.

the development of software products. On the vertical axis, the potential participants are listed, while the shading indicates the level of involvement for each party at the development phase.

The *Multimedia* CEO couldn't quite well illustrate when and how much each function was involved in the phases of development. They had processed specifications only from the software engineering perspective. Rather, he pointed out the distinct role of the management team to take care of the business, software engineers to develop and salesman to deliver the product to customers. Despite the lack of structure, this close-knit management team was effective in communication and decision-making. However, the management group was heavily involved in up front phases, as this decision-making became irregular in the later phases. There were no coordinated activities to deliver the product to the market. On the contrary, the head of business development of *Messaging* indicated the roles of each participant in the development phases without difficulties. They had detailed process descriptions with the personal accountabilities specifying the outcomes of the phases. After all, concerning both firms, this analysis corresponded well with the first analysis thus corroborating and augmenting evidence of this study.

Conceptual model of software product development

The preliminary conceptual model is illustrated in Figure 4 presenting the phases and the identified communication flows. The model synthesizes our findings so far. The information flows originated in the NPD approach and the IS methods are depicted with solid arrows. These represent a fairly linear flow, where information is enhanced during the process. An exception was the middle part, where the respondents described the implementation and component factory to be iterative. Finally, the process ends with the launch of the product.

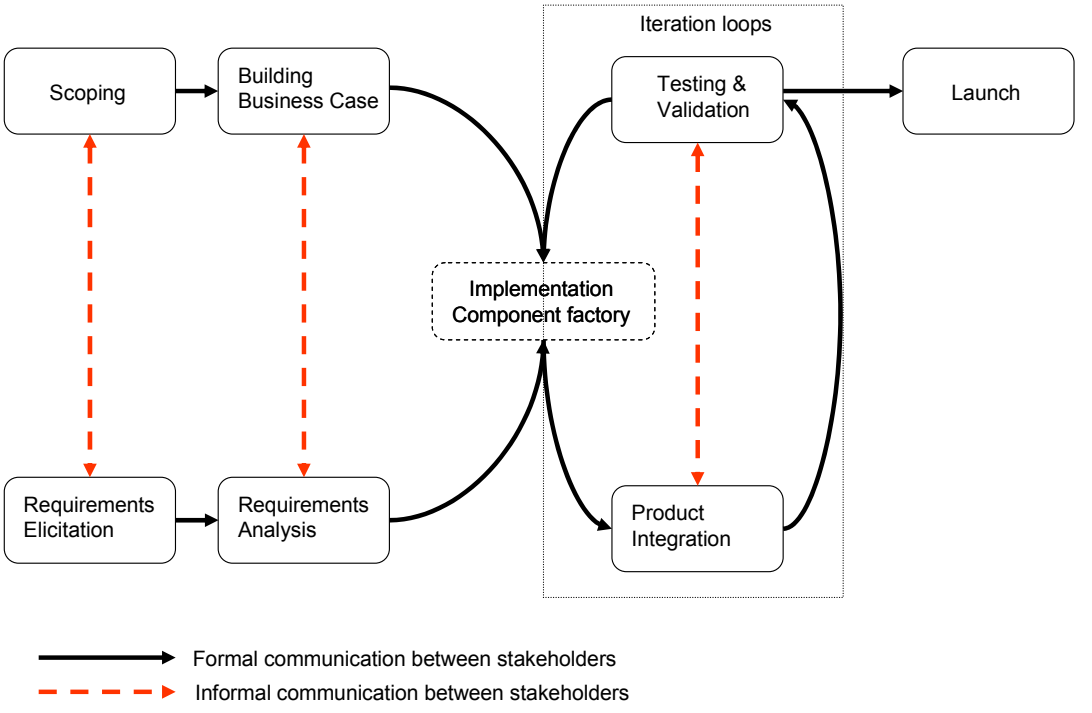

Figure 4. Communication flows in software product development

The four red arrows point out the main communication flows between the two processes. At those points, the teams got meaningful, high-fidelity feedback on the performance of the product and undertook responding to that information. By doing this, the firms identified the main obstacles that needed to work on to get the process to be more flexible. However, it often remained tacit how the information was exchanged between the different participants. Three points in which the participants sensed and responded to changing requirements, are represented by the three dashed arrows.

The first link connected the scoping to the requirements elicitation. In both cases, at this point the necessary information was collected to define "What will be built?" The information such as customer needs and requirements, market development, technology, competitor follow-up and development proposals were gathered trough. These details were gathered mainly in informal discussions. In both firms, the management group confirmed the definitions before moving on to the next stage. At the second point, a software product was aligned with the business strategy. Therefore, all main functions were assigned to verify the business case and look for feedback from their environment. Although *Messaging* committed a somewhat deeper analysis, there were no major differences between the two firms regarding this point. At the third point, the emphasis was on integration. Development and validation activities were carried out concurrently with rapid feedback across these activities. At this point, near final versions of the components of the product were used in both firms as prototypes. As software engineers were correcting some remaining defects, customer trials and test marketing were conducted with the prototypes. Occasionally, the product was introduced to the markets with partners.

DISCUSSION

This study focuses on the development of software products. Our purpose was to demonstrate the integration of the NPD approach with the IS development methods and show its usefulness in developing the responsiveness of software product development to market opportunities.

The findings suggest that development of mobile software products is distinctly divided into two parts, according to how it deals with project risks. We contend that the evolutionary NPD stream, based on the staged approach (McConnell 1986), focuses on managing business risks, whereas the iterative and incremental IS development process places more emphasis on technological risks. However, the two processes were highly intertwined through an informal, intensive and rich communication during development, and through cross-functional teams. The two firms had successfully adjusted the staged product development to accommodate the iterative IS development methods (Beck 1999a; Beck 1999b; Kruchten 1996) for providing possibilities for striking a fine balance between flexibility and control. We found that executives valued the emphasis on up-front business planning and late-phase product launch preparation. Managers used iteration-end dates as milestones to gain a better control over schedule and scope. Developers, in turn, favored the iterative and incremental way of doing things when working towards a solution during implementation, while testers would begin their work earlier in the process, thus being able to identify defects early.

Furthermore, the findings show that the cross-functional teams are useful in integrating the stream of market information with overall development effort (Barczak 1995; Griffin 1997; Souder 1988). In both firms, the management group was interactive during the development and made decisions in real-time. In addition, at *Messaging*, the product steering group was constantly negotiating how to align internal perspectives with emerging needs and trends in the environment. This also supports the findings of the IS researchers that have promoted enabling communication among different participants in development teams (Curtis, Kellner and Over 1992; Keil and Carmel 1995; Tuunanen 2003). In the traditional IS development literature, the product requirements are commonly developed in the up-front phases, whereas in the two firms the decision-makers were involved throughout all the phases of development (Royce 1970; Sommerville 2001). The cross-functional groups clearly produced more diversified knowledge before and especially during the development, which is in line with the respective literature (Calantone, Vickery and Droge 1995; Cooper 2000).

Moreover, our findings imply that the qualities of integration, communication and flexibility can be embraced by the concept of relationship. Successful product development involves forging and nurturing relationships between the different operations involved in product development, between customers and suppliers, and between joint technology partners. For example, *Messaging* used several partnerships to enhance their responsiveness to new market opportunities. It seems that small firms can enhance the market responsiveness of the software product development by fostering a culture of communication and integration. Furthermore, this communication should not be limited to a firm but to justify the innovation process through a continuously evolving discourse of the various external parties involved (Hagedoorn 1993; Nambisan 2002).

The results show that encouraging communication during the development of software products can help to strengthen the degree of integration between different functions. In the two firms, the new information received by the teams was mainly generated and responded to at the three identified connector points. The empirical results indicate that this communication can be supported by visualization and simulation, e.g. through prototyping. This proved to help navigate through a series of decisions on the way and was, therefore, likely to improve the responsiveness to the market. Contrarily to the traditional approach where prototyping is used primarily as an engineering tool for managing technical risks and test design feasibilities (Smith 2001; Sommerville 2001), feedback was gathered from the external environment in order to adjust plans according to the gained information. In addition to managing technical risks, it appeared that the process was also built to manage business risks with regular customer feedback loops. In particular, the findings point out the importance of informal, tacit communication as a basis for these interactions.

The study also helps understand differences at the organizational level. For instance, although

the two firms had quite similar resources, they had different history of assessing and innovating software product development. Because of longer history, *Messaging* had been able to evaluate and reflect on their performance against past development efforts. This experience helped to develop systemic and coordinated communication. On the other hand, *Multimedia* had recently peaked and become more complex, and they had only loose processes in place to facilitate the integration of market knowledge with their domain. Thus, bringing together different sources of expertise was performed in an ad hoc fashion.

CONCLUSION

This study focuses on the development of mobile software products in small software firms. Based on our literature review, we present that the current IS development methods fail to incorporate market elements into the development of software products. We propose that the marketing-related NPD discussion provides the IS discipline with valuable insights into software product development and strengthens the focus on the market opportunities. The objective of NPD systems is to create new, successful product designs, while the IS methods aim at producing better and more predictable results, and implementing those improvements in the functioning of an organization. The former focuses on business risks, such as information about business objectives, customers, competitive environment and the alignment with internal functions, while the latter is strongly based on the contextual knowledge of how information systems and the latest technological developments can be used for the benefit of a customer.

In an interpretive case study of two mobile software firms, we applied both the NPD approach and the IS development methods as a lenses to identify how the participants were involved in the development of software products, and how the information was communicated between them during the phases of development. The applied framework helped us form a coherent view of the development of software products, and it showed preliminary results how to improve software product development to respond to market opportunities. As a synthesis of this study, a preliminary conceptual model is also presented describing the communication flows during the development of software product development.

While more empirical work is necessary to elaborate and verify our results, we believe that a useful starting point has been made. The purpose of our case selection was to obtain information from comparable cases, which adds confidence and robustness to the findings, but it does not ensure generalizable results. Empirical validation of the chosen approach in other settings is clearly needed. The theoretical framework was applied only to two sites, albeit in-depth. More empirical grounding will sharpen and enrich the understanding of the integration of the NPD with the IS methods and yield more complex insight to the development of software products.

In the future, we are seeking to study the recognized informal communication flows in software product development in more depth. Action research (Iversen, Mathiassen and Nielsen 2004) could provide interesting research agenda for a better understanding of these information flows. Furthermore, as organizations grow, they will most likely no longer be able to co-locate their important team members, because the products are developed across boundaries, time zones, and enterprises. This creates challenges for helping virtual product development teams to deliver successful results. Another interesting aspect that requires further investigation is concerned with the question if knowledge management capabilities can be used for distinguishing firms with a lesser success potential.

REFERENCES

Abrahamsson, P., "Extreme Programming: First Results from a Controlled Case Study," *Proceedings of the in the Proceedings of the Euromicro 2003*, Antalya, Turkey, 2003,

Abrahamsson, P., J. Warsta, M.T. Siponen and J. Ronkainen, "New Directions on Agile Methods: A Comparative Analysis," *Proceedings of the in the Proceedings of the 25th International Conference on Software Engineering*, Portland, Oregon, 2003, pp. 244-254.

Alter, S. and M. Ginzberg, "Managing uncertainty in MIS implementation," *Sloan Management Review*, 1978, 20: 1, pp. 23-31.

Barczak, G., "New Product Strategy, Structure, Process, and Performance in the Telecommunications Industry," *Journal of Product Innovation Management*, 1995, 12: 3, pp. 224.

Barki, H., S. Rivard and J. Talbot, "Toward an Assessment of Software Development Risk," *Journal of Management Information Systems*, 1993, 10: 2, pp. 203-225.

Barley, S.R., "Technology as an Occasion for Structuring: Evidence from Observations of CT Scanners and the Social Order of Radiology Departments," *Administrative Science Quarterly*, 1986, 31, pp. 78-108.

Beck, K., "Embracing Change with Extreme Programming," *IEEE Computer*, 1999a, 32: 10, pp. 70-77.

Beck, K., *Extreme Programming Explained: Embrace Change*, Addison-Wesley Longman Inc., Reading, MA., 1999b.

Beck, K., M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R.C. Martin, S. Mellor, K. Schwaber, J. Sutherland and D. Thomas, "Manifesto for Agile Software Development," 2001. Available at: http://www.agilemanifesto.org/, last accessed 26th January 2006.

Boehm, B., "A Spiral model of software development and enhancement," *IEEE Computer*, 1988, 21: 5, pp. 61-72.

Boehm, B.W., B. Clark and e. al., "Cost model for future life cycle processes: COCOMO 2," *Annals of Software Engineering*, 1995, 1, pp. 57-94.

Calantone, R.J., S.K. Vickery and C. Droge, "Business Performance and Strategic New Product Development Activities: An Empirical Investigation," *Journal of Product Innovation Management*, 1995, 12: 3, pp. 214-223.

Chiesa, V., P. Coughlan and C.A. Voss, "Development of a Technical Innovation Audit," *Journal of Product Innovation Management*, 1996, 13: 2, pp. 105-136.

Clark, K.B. and S.C. Wheelwright, *Managing New Product and Process Development: Text and Cases*, The Free Press, New York, 1993.

Cockburn, A., *Agile Software Development*, Addison-Wesley, Boston, 2002.

Cooper, R., "The New Product Process: An Empirically-Based Classification Scheme," *R&D Management*, 1983, 13: 1, pp. 1-13.

Cooper, R., "Winning with New Products Doing it Right," *IVEY Business Journal*, 2000, July-August 2000, pp. 54-60.

Cooper, R.G., "Stage-Gate Systems - a New Tool for Managing New Products," *Business Horizons*, 1990, 33: 3, pp. 44-54.

Curtis, B., M.I. Kellner and J. Over, "Process modeling," *Communications of the ACM*, 1992, 35: 9, pp. 75-90.

Davis, G., "Strategies for information requirements determination," *IBM Systems Journal*, 1982, 21: 1, pp. 4-31.

De Maio, A., R. Verganti and M. Corso, "A Multi-Project Management Framework for New Product Development," *European Journal of Operational Research*, 1994, 78: 2, pp. 178-191.

Dove, R., Response Ability: The Language, Structure, and Culture of the Agile Enterprise, John Wiley & Sons, Inc., New York, 2001.

Eisenhardt, K.M., "Building Theories from Case Study Research," *Academy of Management Review*, 1989, 14: 4, pp. 532-550.

Glaser, B. and A. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine Publishing Company, Chicago, 1967.

Greiner, L.E., "Evolution and Revolution as Organizations Grow," *Harvard Business Review*, 1998,

Griffin, A., "PDMA Research on New Product Development Practices: Updating Trends and Benchmarking Best Practices," *Journal of Product Innovation Management*, 1997, 14: 6, pp. 429-458.

Gupta, A., "The R&D-Marketing Interface in High-Technology Firms," *Journal of Product Innovation Management*, 1985, 2: 1, pp. 12.

Haeckel, S.H., "Adaptive Enterprise Design: The Sense-and-Respond Model," *Planning Review*, 1995, 23: 3, pp. 6-42.

Hagedoorn, J., "Understanding the Rationale of Strategic Technology Partnering: Interorganizational Modes of Cooperation and Sectorial Differences," *Strategic Management Journal*, 1993, 14, pp. 371-385.

Hart, S.J. and M.J. Baker, "The Multiple Convergent Processing Model of New Product Development," *International Marketing Review*, 1994, 11: 1, pp. 77-92.

Hauser, J.R., "The House of Quality," *Harvard Business Review*, 1988, 66: 3, pp. 63-73.

Heaton, L., "Talking Heads vs. Virtual Workspaces: A Comparison of Design across Cultures," *Journal of Information Technology*, 1998, 13, pp. 259-272.

Highsmith, J. and A. Cockburn, "Agile Software Development: The Business of Innovation," *Computer*, 2001, 34: 9, pp. 120-122.

Iversen, J.H., L. Mathiassen and P.A. Nielsen, "Managing Risk in Software Process Improvement: An Action Research Approach," *MIS Quarterly*, 2004, 28: 3, pp. 395-433.

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

Klein, H.K. and M.D. Myers, "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems," *MIS Quarterly*, 1999, 23: 1, pp. 67-93.

Kruchten, P., "A Rational Development Process," *Crosstalk*, 1996, 9: 7, pp. 11-16.

Lee, A.S., "A Scientific Methodology for MIS Case Studies," *MIS Quarterly*, 1989, 13: 1, pp. 33-52.

Mata, F.J., W.L. Fuerst and J.B. Barney, "Information Technology and Sustained Competitive Advantage: A Resource-Based Analysis," *Mis Quarterly*, 1995, 19: 4, pp. 487-506.

Mathiassen, L., T. Saarinen, T. Tuunanen and M. Rossi, "Managing Requirements Engineering Risks: An Analysis and Synthesis of the Literature," Helsinki School of Economics, November 2004, pp. 63.

McConnell, S., "Rapid Development-Taming wild software schedules," 1986,

McFarlan, F.W., "Portfolio Approach to Information Systems," *Harvard Business Review*, 1981, 59: 5, pp. 142-150.

McGrath, M.E., Setting the PACE in Product Development: A Guide to Product and Cycle-Time Excellence, Butterworth Heinemann, 1996.

Merisalo-Rantanen, H., T. Tuunanen and M. Rossi, "Is Extreme Programming Just Old Wine in New Bottles: A Comparison of Two Cases," *Journal of Database Management*, 2005, 16: 4, pp. 41-61.

Merton, R.K., M. Fiske and P.L. Kendall, *The Focused Interview: A Manual of Problems and Procedures*, Free Press, New York, 1990.

Miles, M.B. and A.M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook*, Sage Publications, Newbury Park, CA, 1994.

Nambisan, S., "Complementary Product Integration by High-Technology Ventures: The Role of Initial Technology Strategy," *Management Science*, 2002, 48: 3, pp. 382-398.

Nandhakumar, J. and J. Avison, "The Fiction of Methodological Development: A Field Study of Information Systems Development," *Information Technology & People*, 1999, 12: 2, pp. 176-191.

Orlikowski, W.J., "CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development," *MIS Quarterly*, 1993, 17: 3, pp. 309-340.

Orlikowski, W.J. and J.J. Baroudi, "Studying Information Technology in Organizations: Research Approaches and Assumptions," *Information Systems Research*, 1991, 2: 1, pp. 1-28.

Pressman, R., Software engineering -a practitioners approach, McGraw-Hill, 2000.

Regnell, B., M. Hösta, J.N.o. Dag, P. Beremark and T. Hjelm, "An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software," *Requirements Engineering Journal*, 2001: 6, pp. 51-62.

Royce, W.W., "Managing the development of large software systems: concepts and techniques," *Proc. IEEE WESTCON*, 1970, pp. (Ch. 3).

Rubin, H.J. and I.S. Rubin, *Qualitative Interviewing: The Art of Hearing Data*, Sage, Thousand Oaks, CA, 1995.

Smith, P.G., *Time-Compression Technologies*, 2001, April 2001, pp. 18-24.

Sommerville, I., *Software Engineering*, Addison-Wesley, U.S., 2001.

Souder, W.E., "Managing Relations Between R&D and Marketing in New Product Development," *Journal of Product Innovation Management*, 1988, 5, pp. 6-19.

Truex, D., R. Baskerville and J. Travis, "A Methodical Systems Development: The Deferred Meaning of Systems Development Methods," *Accounting, Management and Information Technologies*, 2000, 10: 1, pp. 53-79.

Tuunanen, T., "A New Perspective on Requirements Elicitation Methods," *JITTA: Journal of Information Technology Theory & Application*, 2003, 5: 3, pp. 45-62.

Ulrich, K.T. and S.D. Eppinger, *Product Design and Development*, The McGraw-Hill Companies, Inc., Boston, 2000.

Vainio, A.M., T. Tuunanen and P. Abrahamsson, "Developing Software Products for Mobile Markets: Need for Rethinking Development Models and Practices," *Proceedings of the Hawai'i International Conference on System Sciences HICSS-38*, Hawaii, the Big Island, 2004, pp. 10.

Walsham, G., Interpreting Information Systems in Organizations, Wiley & Sons, Chichester, 1993.

Yin, R.K., *Case Study Research: Design and Methods*, Sage Publications, Inc., Thousand Oaks, 1994.

Zahay, D., A. Griffin and E. Fredericks, "Sources, Uses, and Forms of Data in the New Product Development Process," *Industrial Marketing Management*, 2004, 33: 7, pp. 657-666.

APPENDIX 1 INTERVIEWS NOTES FOR CROSS-CASE ANALYSIS

The participants were identified in the interactions of software product development. In the table, the quotations collected for the analysis are presented

Stages	Multimedia	Messaging
Scoping / Requirement Elicitation	The CEO: "The management group focuses on what will be built and tries to collect information through our industry network at customers, partners and technology forums to make a prelim market assessment." The CEO: "We have formed few partnerships with telecom operators just to be more alert in the market. They aren't the most profitable customers, but they will inform us of the future development." The CEO: "We don't have any link to the end users. It would be too expensive us to collect systematically data from the market." The CEO: "We know a lot of people in this industry and we talk a lot with them. Of course, our salesmen are able to collect a lot of feedback."	The head of business development: "Account managers collect feedback from customers and save it in a database." Competitors are tracked systematically with the help of supply requests and market follow-up. Just yesterday I finished a slide show about competitors." The CEO: "We collect customer feedback through a digital channel and stored it in a database. The responses are given priorities according to three levels and the results are reviewed in the management group and the product steering group." The head of business development: "Last year we conducted a market study for the main customers. We wanted to study the current market situation, the future trends, and the timing of those trends. On the basis of the results, we adjusted our plans. We [marketing department] will repeat the research yearly." The CTO: "The quality survey is conducted for current customers at regular intervals."
Building Business Case / Requirement Analysis	The CEO: "However, because of limited resources, we have to trust on the common touch. Last time the management group held a meeting in a cottage in Lapland. We didn't come back until we had finalized the plans." The CEO: "All [developers and account managers] talk a lot around the coffee table to analyze the plans." The head of technology: "After defining an initial value proposition, the salesmen of the firm start to collect the final proofs by questioning few customers: Would you like this kind of feature..?"	The CEO: "All functions perform analysis in traditional ways. For example, financial calculations are very important." The head of business development: "The decisions are reviewed by the product steering group that gathers all functions together. We meet once in three months and discuss the main guidelines for the new product development." The CTO: "Software engineers contribute their part by testing the concept of an idea. Then we simply tell the idea to our customers and let them play with demos. If they would be willing to pay for the idea, then we let thighs slide." The CEO: "The product steering group decides when to start the development and the management group approves the definitions and plans."

The participants were identified in the interactions of software product development. In the table, the quotations collected for the analysis are presented (Cont'd)

Implementation / Component Factory	The head of technology: "The product features are developed according the plans. The development team is responsible for the technological feasibility of the product concept. "	The CTO: "Based on the plans, software engineers and technology architects start development. It is better that there are different people to figure out what is the easiest way to develop features which is not the point on previous stages. Their role is sort of to optimize the plans."
Testing & Validation / Product Integration	The CEO: "A prototype is always developed, but it actually serves more the sales department than development. It is not productive to present a piece of code to anyone; we have to conceptualize the product to collect the final customer feedback. Otherwise, it is difficult to transmit an image of the product concept. At this point, the sales and business development manager start to contact potential customers." The head of technology: "First, the functionality of the product is verified internally [by software engineers]. Then, external testing of the product feature is performed by delivered demo versions to customers in order to get feedback. However, this requires the same amount of work than normal product implementation why we don't perform this in every case.	The CTO: "Software engineers test the product functionality and then perform customer field trials. I and some account managers discuss with customers to get the in-depth feedback. The customers explore the product from the business perspective." The CEO: "The head of business development with marketing contacts potential sales partners. They strengthen our position because then we can offer a comprehensive solution."
Launch	The CEO: "Market launch and roll-out is often performed in the most important fairs, such as Cannes and GSM World. I go there with sales and marketing. The purpose is to present the product to customers." The head of technology: "Based on the first feedback, we might still fix some technological problems."	The CEO: "Market introduction is performed at the main industry fairs, such as GSM World or Cannes. All main functions are present, but, of course, sales and marketing are doing the main job meeting customers." The CEO: "The product is also introduced to our partners and to members of technology forums." The CEO: "Results are monitored in the management group. And then there are that after sales quality survey.."

AUTHORS

Tuure Tuunanen, D.Sc. (Econ) (Helsinki, 2005), received his doctoral degree in information systems science at the Helsinki School of Economics. His current research interests lie in the area of IS development methods and processes, requirements engineering, and convergence of IS and marketing disciplines in design. He is currently senior lecturer at University of Auckland Business School. His research has been published in Information & Management, Journal of Database Management, Journal of Information Technology Theory and Application, and Journal of Management Information Systems. In addition, his work has appeared in a variation of conference proceedings within his research interest areas, such as eCOMO, DESRIS, ISD, HICSS, Mobility Roundtable, RE, WeB, and WITS. Up-to-date information about his research is available at http://www.tuunanen.fi.

Anu Marianne Vainio, M.Sc. (Econ) (Jyväskylä, 1999), is pursuing a D.Sc. degree in information systems science at the Helsinki School of Economics. Her research focuses on coordination of dynamic capabilities during software product development in small firms. Her research has been published in *European Journal of Marketing* and in several international conferences. Furthermore, she has extensive experience in the areas of financial management, strategic planning and project management in software industry. Marianne Vainio is currently working as a senior advisor at Information Risk Management, KPMG Finland.