Asymmetries of Knowledge Between Engineering and Marketing in Software Product Development

Mikko Ronko
Turku Centre for Computer Science, mikko.ronko@utu.fi

Markus Makela
University of Turku, markus.makela@utu.fi

Follow this and additional works at: http://aisel.aisnet.org/ecis2008

Recommended Citation
http://aisel.aisnet.org/ecis2008/223

This material is brought to you by the European Conference on Information Systems (ECIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2008 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
ASYMMETRIES OF KNOWLEDGE BETWEEN ENGINEERING AND MARKETING IN SOFTWARE PRODUCT DEVELOPMENT

Rönkkö, Mikko, Turku Centre for Computer Science, Joukahaisentie 3-5, 20520 Turku, Finland, mikko.ronkko@utu.fi
Mäkelä, Markus M., University of Turku, Joukahaisentie 3-5, 20520 Turku, Finland, markus.makela@utu.fi

Abstract

This paper reports the results of an inductive case study about asymmetries of knowledge between marketing and engineering subunits in firms engaged in software product development. The asymmetries of technical and user related knowledge are salient to software since the artifact itself is invisible and complex, and have been previously reported as a key challenge in product oriented software development.

We use qualitative interview data from six firms using multiple informant design. The analysis follows the constant comparison method where the initial constructs were defined ex ante, but their exact relationship and the moderating constructs were derived from the data.

We found that not only is there a strong asymmetry of what the focal units know, but that this has an effect on software product development performance. Our analysis suggests that the negative effects of knowledge asymmetry are negatively moderated by presence of a strong boundary spanning individual and also suppressed by organizational change.

Keywords: Knowledge, asymmetries of knowledge, shared knowledge, software development, product development.
1 INTRODUCTION

Developing software products requires collaboration between several functional units of a firm (Karlsson, Dahstedt, Regnell, Dag and Persson, 2007). In this paper, we examine the effect of knowledge asymmetries – the effect of uneven distribution of different types of knowledge - between different functional units and what impact this phenomenon has on software product development performance. The significance of effects of shared knowledge and distributed knowledge on product development has been documented in the product development (Ottum and Moore, 1997) and management (Hoopes and Postrel, 1999) literature, but knowledge as a general topic remains largely under-explored in the context of software development (Bjornson and Dingsoyr, In Press).

Software development provides a fruitful context for studying knowledge sharing and cross-functional integration, since the intangibility and complexity of software creates challenges for sensemaking (Sonnentag, 1998; Weick, 1995). That is, the complexity of the technology tends to make the internal structures of systems less observable, and intangibility prevents the artefact from providing any visible cues that would help people to understand what happens beneath the user interface. As a consequence, when technology is complex, the perception of structure is strongly affected by the interactions that the focal person has with the technological artefacts. For example, in the case of information systems, it is often the features, not structure that acts as a trigger for sensemaking in the case of users (Griffith, 1999). In contrast, engineers often tend to form their own mental models based on highly technical details (Sonnentag, 1998).

The previous research in the interdepartmental sensemaking in new product development has mainly focused on traditional and stable industries. However, several sources provide evidence that organizations that operate within more turbulent environments and develop more complex technology have different sensemaking processes than in their counterparts in less volatile industries (Bogner and Barr, 2000; Dougherty, Borrelli, Munir and O'Sullivan, 2000). Clearly, this indicates that the theories constructed using data from more traditional industries should be complemented by studies in the context of the software industry.

This paper presents the results of a theory building multiple case study of six Finnish software product firms. The rest of the paper is structured as follows: We will first introduce the key constructs of the paper, as defined prior to data collection, and review the relevant literature. The section thereafter describes the data and the empirical study design. The main part of the paper consists of presenting summary results of the analysis providing a clear link between the results and data. Finally, we conclude by evaluating the results and presenting suggestions for future research.

2 KEY CONSTRUCTS AND LITERATURE REVIEW

In qualitative research, the researcher often strives to develop theory based on data without any a priori assumptions of the key constructs, but this is neither necessary nor always desirable (Eisenhardt, 1989). In this paper we build a theory about the causal dependency between inter-functional knowledge asymmetries and product development performance. Based on the literature review, we hypothesise that knowledge asymmetries do have a causal relationship with product development performance. However, while several effects of asymmetries of knowledge have been reported, the exact nature and the role of these asymmetries on software product development are not yet known. Due to this, the present paper adopts an inductive theory building research design.

2.1 Knowledge asymmetries

While several different theories explain different types of knowing in organizations, we still lack a general theory in the area (Fiol, 2002). It is generally agreed that one of the strengths of a modern
organization is that it can know a lot more than an individual member of the organization (Grant, 1996; Kogut and Zander, 1992; Spender and Grant, 1996), and the knowledge of the organization is not necessarily tied to individuals, but tends to be sticky even when people change (Brown and Duguid, 2001). There is an agreement that knowledge resides in routines (Nelson and Winter, 1982), network ties and coordination (Kogut and Zander, 1996), and technology (Orlikowski, 1992). However, understanding how an organization can accomplish task demanding complex knowledge is only part of a bigger picture of knowledge. While the procedural knowledge is embedded in the organization and routines, semantic knowledge, or establishing of meaning can be considered as an emergent accomplishment through the ongoing process of sensemaking where people interact and socialize to create meaning for different events and cues (Weick, 1995).

We define knowledge asymmetry as a condition that is created when different people or different organizational units possess different stocks of knowledge. In other words, knowledge asymmetry is the result of the uneven dispersion of different types of knowledge within the organization (Becker, 2001). Since knowledge is accumulated over time as result of experiences, and different positions inside a firm are exposed to different types of experiences, it is inevitable that knowledge is not symmetrically distributed within a firm. However, the degree of asymmetries, or the relative lack of common ground, can change from one organization to another.

The stock of knowledge possessed by a person affects how they give meaning to different information and situations causing people to interpret their observations in a way that fits their belief structure (Gioia and Poole, 1984). In the context of information systems and software development, people with technical and non-technical backgrounds view systems under development often quite differently. These differences are often both a product and a cause of emergence of two separate communities of practice (Bechky, 2003) within a software firm: the marketing people, and the engineering people. While engineers tend to make sense of the technology and structure their knowledge through the technical architecture (Sonnentag, 1998; von Mayrhauser and Vans, 1995), the non-technical people in the marketing team think about the technology through features and user interface (Griffith, 1999; Orlikowski, 2000).

In all but the smallest firms, there is an organizational boundary between these two focal units, partially caused by the knowledge asymmetries due to different backgrounds and different functional role within the firm. It is generally harder to communicate over the boundary and progressively harder as the boundary widens (Carlile, 2004). As a workaround to the problem of knowledge asymmetries, organizations often tacitly implement transactional memory systems, where the organizational knowledge is a sum of individuals’ knowledge and collective awareness of who knows what (Austin, 2003). However, these systems can only partially solve the problem of asymmetric distribution of knowledge, and communications over disciplinary boundaries and combining market and technological knowledge remain problematic in software product development (Karlsson et al., 2007).

2.2 Product development performance

Product development performance can be considered to be a three dimensional construct with effectiveness, efficiency and innovativeness as sub-dimensions (Kusunoki, Nonaka and Nagata, 1998): Efficiency refers to the goodness of the rate that the resources are transformed into outputs. This dimension has a close fit to the process performance dimension of software development performance (Jiang, Klein, Hwang, Huang and Hung, 2004). Effectiveness refers to the goodness of the product not only in terms of product quality – a common measurement of software development performance (Jiang et al., 2004) – but also how well it fits the needs of the markets. Finally, innovativeness refers to the ability conceive new ideas and develop these into commercially successful products or product features.

Asymmetries of knowledge affect can affect the product development performance through several different mechanisms. Generally, asymmetries of knowledge hinder transfer of information relevant
for the development process. For example, the asymmetry can increase the transaction costs of communication hence decreasing the amount of information sharing (Carlile, 2004), cause bias for using the knowledge that is shared and omitting what is not shared (Gigone and Hastie, 1993; Stasser, Stewart and Wittenbaum, 1995). This can affect product development performance in several ways: First, lack of knowledge can cause ‘glitches’, that is, developing something that is not what was intended (Hoopes and Postrel, 1999). Second, knowledge asymmetries can cause problems to be undetected (Cramton, 2001). Third, asymmetries hinder effective coordination of work (Faraj and Sproull, 2000).

However, asymmetries of knowledge can also have a positive impact on software product development performance. Previous research has identified that innovation, defined as creating something novel and useful, occurs when knowledge from different domains is combined and processed (Carlile, 2004; Kogut and Zander, 1996; Marsh and Stock, 2006). If the degree of knowledge asymmetries is low, this is likely to lead to reduced creative abrasion and hence decreased innovativeness (Leonard-Barton, 1995).

Clearly, there is not a lack of theories considering the knowledge asymmetries and their effect on product development performance. However, the current literature is not clear about the overall net effect of knowledge asymmetries on software product development performance, calling for an inductive study examining the nature of the relationship between these two constructs.

3 DATA AND METHODS

3.1 Research strategy

This paper follows theory building multiple case study approach (Eisenhardt, 1989). More specifically we follow the “variance study” design (Van de Ven, 2007). In this approach, novel theory is formulated as a result of thorough and disciplined analysis of empirical material in such a way that the researcher goes to study the data, at first, in an exploratory manner and then discovers causal ties from it, making comparisons with existing literature. Our approach to this research follows that of Eisenhardt (1989) even though it has commonalities with those described by Strauss and Corbin (1998) and Yin (2003) in more recent literature. In this method, multiple cases are selected for study due to the replication and thus a stronger claim on validity that they enable (Eisenhardt, 1989; Yin, 2003).

3.2 Sample

Our sampling strategy follows theoretical sampling as described below. Data for this paper was collected from eight firms, but two cases were later dropped due to achieving theoretical saturation (Strauss and Corbin, 1998) without the need for further data collection warranting finalizing data collection for these cases. Since this study is about interdepartmental knowledge asymmetries, we focus on firms where interdepartmental communication is rich to gain understanding of the differences in semantics of the knowledge (Carlile, 2004) between departments. We chose smaller firms for two reasons: First, smaller firms rely more on interpersonal coordination mechanisms while larger firms build processes and routines (Nelson and Winter, 1982). Due to the nature of product development capability as a dynamic capability, these best practice routines often emerge and spread across firms as best practices (Eisenhardt and Martin, 2000). Due to this, the variance between firms was expected to be higher in smaller firms that rely on interpersonal coordination than larger firms that rely predominantly on routines. Second, Finnish software industry is characterized by a large number of small firms, and hence this was also a choice of convenience.

Our study design requires sampling firms, which have clear division of labour and clear internal team structure limiting the size of the firms included in the population where the sample is drawn, and
hence we chose 20 people to be a minimum size for a firm to be included in the study. Similarly, we did not want to include highly entrepreneurial start-up firms for several reasons: First, these firms often have not institutionalized processes and organizational structures, and would not be comparable with more mature firms. Second, entrepreneurial firms are often championed by the entrepreneur, which significantly affects the sensemaking processes of the organization (Hill and Levenhagen, 1995). Third, start-up firms in the Finnish software industry are often technology driven, and hence they might not balance between market and technology orientation as a more mature firm does (Drazin, Glynn and Kazanjian, 1999). The minimum age was set at 5 years. Due to practical reasons, the population was limited to firms whose head office was located in Southern Finland.

A sample of eight firms was drawn from along two dimensions: environmental turbulence and product complexity. The first dimension, turbulence, was chosen because existing research indicates that organizations who operate in turbulent environments have different sensemaking processes and adopt different organizational structures than organizations within less turbulent environments (Bogner and Barr, 2000; Calantone, Garcia and Droge, 2003). The second dimension was chosen for two reasons, first it was hypothesised by the authors that more complex products will lead to more knowledge asymmetries and second, existing research shows that product complexity affects both organizational design (Sosa, Eppinger and Rowles, 2004) and product development performance (Rajiv, Gordon and Sandra, 1998). We identified the four quadrants of turbulence-complexity matrix as theoretically interesting groups. We chose two firms from each group based on the convenience of the location and contacted these for interview. When data were analyzed, theoretical saturation was achieved before finalizing two cases, and hence we exclude these cases from the analysis. Table 1 presents a summary description of the six cases that were included.

The unit of analysis selected was the software firm. Several types of evidence were used: interviews, observations, and secondary sources such as company websites, press releases, and items from the news media. Altogether 32 interviews provided data for the final analysis Majority of the data was collected during spring 2007, but secondary data was obtained also throughout the entire analysis and writing phase of this paper. All interviews were conducted in physical meetings and all informants provided their permission for taping the interview, and taped interviews were later transcribed for analysis. Extensive field notes were take following the 24 hour rule (Miles and Huberman, 1984). The interviews were semi-structured to allow the conversation to take directions according to the informants' responses, and to allow an in-depth inquiry into the nature of the subject issues.

3.3 Data analysis

The interviews resulted in approximately 500 pages of interview transcripts and field notes with single spacing and 12pt font. The analysis comprised within- and cross-case analyses as outlined by Eisenhardt (1989). This involves, first, the extensive study of each case individually, treating it in the way of an “experiment,” and typing in depth case descriptions of each firm. Second, similarities and differences in two or more cases were sought such that the emergence of new categories and concepts is facilitated. These differences were identified by inductively seeking patterns in the data, and abductively by seeking plausible explanations for observed variations of outcomes. Once identified, these patterns were approached deductively as to identify if the effect was universal in the sample, and where needed the emergent theory was revised or rejected. In this iterative manner rotation between data, emerging theory, and our knowledge of prior literature were employed in the analysis process. Tables were used to present the variance in constructs and as aids when searching for patterns in data (Eisenhardt, 1989; Miles and Huberman, 1984). Data analysis was performed with the help of NVivo 2.0 qualitative data analysis software.

Triangulation was made use of by using several informants for each case to reduce some of the possible biases in the data, such as those caused by potential ex post rationalization. As is typical for qualitative research in general (Eisenhardt, 1989), the analysis phase overlapped the data collection phase.
<table>
<thead>
<tr>
<th>Firm</th>
<th>Summary description</th>
<th>Product</th>
<th>Organization</th>
<th>Informants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>Alpha has its roots at the late 80's when it was started as a subunit of a larger company. Currently the firm has around 30 employees and can be considered a mature firm.</td>
<td>An information systems product for public sector. The product always requires an implementation project and training.</td>
<td>The company has three offices in different regions with geographically dispersed product development. Three main organizational units are product development, sales and marketing, and services. One development team is off-shored.</td>
<td>CEO, CTO, COO, PM, P, CTO, CS, PM (technical), PM (marketing)</td>
</tr>
<tr>
<td>Bravo</td>
<td>Bravo was started the beginning of this decade. The company is currently growing fast and had approximately 30 employees at the time majority of the interviews were conducted. Bravo received first round of venture capital funding during the interview period.</td>
<td>A data mining and network analysis tool to be used with information systems that contain transaction data. The product development was started at two years go and version 1.0 was released during the interview period. The product can be considered to be new to the world innovative.</td>
<td>The firm has one office and three organizational units related to the development of the system. In addition to characteristic engineering and marketing departments, the firm has a science department which develops the complex mathematical algorithms used in the product. The firm has shared product manager position with two employees.</td>
<td>CEO, VP, CMO, CTO, SM</td>
</tr>
<tr>
<td>Charlie</td>
<td>Charlie was started during the late 90's. The company was not initially information systems provider, but has gone trough several transformations and restructurings to become one. The firm is now employing roughly 30 people and expects moderate growth.</td>
<td>The company produces technology that is used to build media functionality to intranets and public www-sites. The second generation of the system was expected to be release soon after the interview period replacing the first generation product that suffered from significant architectural decay.</td>
<td>The firm is now employing several generations. Delta is a partially owned subsidiary of an engineering office. has its roots in the mid 70's when the development of the current product was started. The company is mature and stable employing 25 people.</td>
<td>CEO, CTO, CMO, COO, SM</td>
</tr>
<tr>
<td>Delta</td>
<td>Delta is a partially owned subsidiary of an engineering office. has its roots in the mid 70's when the development of the current product was started. The company is mature and stable employing 25 people.</td>
<td>Delta produces application software which is used in specialty engineering. The product is currently more than 20 years old, although it has gone through several generations. Delta considers its software to be very complex and highly sophisticated.</td>
<td>The company has three offices in Central Europe and two offices in Finland. Product development is performed in both of these domestic offices. Four organizational units participate in development of the product: engineering, data analysis, and marketing departments, the firm always requires an implementation infrastructure.</td>
<td>CEO, SM, CS, SD, CTO, SM (international), VP (international)</td>
</tr>
<tr>
<td>Echo</td>
<td>Echo is a wholly owned subsidiary of another software firm containing all product business of the parent company. Echo was established at around 2000. At the time of the interviews, the firm employed approximately 30 people mainly in product development. During the interview period, the firm received first round of venture capital funding.</td>
<td>Echo's product is a multiphase communications client that is used in the telecom industry with a variety of third party server systems. The second generation of the product is currently being prepared. The software can be considered to be very innovative and its market is only starting to emerge.</td>
<td>The firm is co-located with the parent company, but there is only little collaboration with the parent. Three organizational units participate in the development: small sales and marketing department, engineering, and customer projects. Product ownership is somewhat vague, and CEO participates actively in development decisions.</td>
<td>P, CEO, CTO, CMO, D, PM</td>
</tr>
<tr>
<td>Golf</td>
<td>Golf is a family owned company which was founded at the 70's. The company is currently planning a transformation to enable international growth. At the time of interviews, the firm employed approximately 40 people. The firm is mature and stable, but is planning for international growth.</td>
<td>Golf produces an information system that is used to facilitate the sales processes and inventory management in a particular industry. Application is provided as ASP. Data from sales transactions are analyzed and fed back to the system for business intelligence purposes. The large role of customer data as a resource has enabled the firm to create a monopoly position that is hard to challenge.</td>
<td>The company has one office in Central Europe and two offices in Finland. Product development is performed in both of these domestic offices. Four organizational units participate in development of the product: engineering, data analysis, services, and sales and marketing. The product ownership is vague, and the CEO actively participates when product development decisions are made.</td>
<td>CEO, SM, CS, SD, CTO, SM (international), VP (international)</td>
</tr>
</tbody>
</table>

Informant positions are indicative. For example, the highest person responsible for technology is termed CTO regardless of the real title. P = President, VP = Vice President, CEO = Chief Executive Officer, CTO = Chief Technology Officer, PM = Product Manager, CMO = Chief Marketing Officer, COO = Chief Operating Officer, CSA = Chief Software Architect, CS = Chief Scientist, CDO= Chief Development Officer, SM = Sales Person, D = Designer, SD= Software Developer

Table 1 Summary description of the cases
4 ANALYSIS AND RESULTS

Tables 2 and Table 3 present the values of the final constructs for the cases with linking to data. During the interviews and data analysis it became obvious that asymmetries in two classes of knowledge were most salient to the development of software products: “I have a technical background so I understand the product, and I have also done customer project so I understand how the software is used” explained Alpha’s product manager. In several other occasions it was also apparent that the most influential asymmetries of knowledge were when product development did not understand the market and use of software or when sales did not understand the technical aspects of the system: “One simple example is digital signing. Our sales people have not necessarily understood that purchasing of the module is not sufficient, but the customer needs to acquire also card readers and another type of server with connections to public network. Also it is very hard to understand what is the difference of publishing information on another system and performing integration.”

Similarly, product development has often less than perfect understanding about the use of the software: “It was a problem with entering data, but I do not remember exactly what it was. Our product development had solved the issue elegantly and technically correctly, but from the customer’s perspective it was a step backwards.” Moreover, it was evident that sales and marketing’s ignorance of the software development process had a negative effect: “Our sales people do not really understand the development process and they want to push new features when we are trying to make stable software.”

On the other hand, it became clear that the too much shared knowledge leads to groupthink and decreasing ability to absorb new information and generate ideas (Jansen, Van den Bosch and Volberda, 2005): “We have worked together so long that everyone knows the software well. I guess we are kind of threading water with this software, everyone is doing their job but not really innovating.”

Moreover, one product manager with a boundary spanning role acknowledged: “When I think the software as a sales person, I consider the customer’s processes and the value that customer gets from our product. But when I switch the engineer’s hat on, I often consider only the technical implementation and what would look elegant … That brings two different views to the issue and this is not necessary a problem, but that both parties can bring their understanding together.” Based on these accounts and the evidence of relationship between asymmetries of knowledge and product development performance presented in Tables 2 and Table 3 we draw the following proposition:

**Proposition 1:** Asymmetries of knowledge have an inverted U-type relationship with product development performance.

One factor seems to affect the effect of asymmetries of knowledge on the product development performance. Charlie had a strong vice president level person with both technical and business background as a strong product champion. Interestingly, in both of the organizations the product owner was officially someone else in the organization. These findings are contradictory to recent study by Ebert (2007), which argued that product manager is often like a mini-CEO, and that product manager is the most important person in development organization. We find that it is not necessarily the product manager, but a product owner or champion who acts as a broker between the different functional units can be highly valuable for a firm. The CTO of Charlie elaborated: “Jack has always the final word on the product. … If I need to know something I always ask Jack”. Similarly, Echo had a product champion with technical and marketing background, although not as strong as Charlie. Neither of these two firms seemed to have any real difficulties caused by asymmetries of knowledge between marketing and engineering. Based on these observations and evidence in Tables 2 and Table 3, we present:

**Proposition 2:** The presence of a strong boundary spanning individual decreases the negative effects of knowledge asymmetries.
The second proposition becomes almost evident when one considers that deciding on information systems requirements is dictated by organizational power (Bergman, King and Lyytinen, 2002). If there is a power full boundary spanning individual he spends time explaining the relevant issues for engineering and marketing in their respective languages and hence acts as a communication link and translator over the pragmatic boundary between the units.

Finally, our data indicates that asymmetries of knowledge can create conflict and cause a feedback loop. In the case of Delta and Golf, the asymmetry had built up and created conflict. When people did not share sufficient common ground, the organization started to feel the negative effects of professionalism. This was caused by institutionalizing of roles and routines and seriously hindered effective decision making in product development decision. The strongest candidate construct for explaining this in our data is organizational change or the lack of that. Hence we propose:

**Proposition 3**: Organizational change negatively moderates (dampens) the effect of negative effects of asymmetries of knowledge.

Contrary to our initial assumptions, we did not find a strong role of environmental turbulence or product complexity on the relationship, as is evident when examining Tables 2 and Table 3.

<table>
<thead>
<tr>
<th>Firm</th>
<th>Description</th>
<th>Asymmetry of knowledge</th>
<th>Product development performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>A lot of the key people interviewed had a technical background and had a relatively long history with the product. The interviewees had little difficulties in describing the more technical aspects of the system.</td>
<td>Low</td>
<td>Not really innovative, but can solve client problems effectively. Some quality problems in the past. Some glitches are reported. Several informants reported that growth could be achieved in the private sector, but it seems that product development has not been able to create a truly compelling feature set for this segment.</td>
</tr>
<tr>
<td>Bravo</td>
<td>There are clearly three different communities of practice within the firm: programmers, algorithm developers and business people. These have some problems understanding each others, but due to mutual trust problems can be solved rapidly.</td>
<td>Med</td>
<td>The product is technologically advanced and technically of high quality. The company considers itself to be highly innovative and customer oriented. Schedule slips due to unrealistic effort estimates seem to be largest problems in development.</td>
</tr>
<tr>
<td>Charlie</td>
<td>Sales and project business department considers the product as a black box and does not seem to have a clear idea of how the product is developed. Moreover, the sales and product development are geographically separated, which increases the difference between these units.</td>
<td>High</td>
<td>The company develops new generations from its product at an astonishing speed. The company seems to be really innovative, and has a decent track record in solving customer problems. However, there have been some technical weaknesses in the products.</td>
</tr>
<tr>
<td>Delta</td>
<td>All people seem to know the product well. There is technical jargon related to the product. But key people have learned to know that. Sales argue that engineering does not understand them and visa versa. Although most of the people have a similar technical background and share a long history, cross-functional integration seems hard to accomplish.</td>
<td>High</td>
<td>The company has been able to survive in the market for more than 20 years. However, currently the product development is not proceeding as fast as the company would desire. Also, there are some problems in staying on schedule and prioritization of different requirements is often done in a non-optimal way.</td>
</tr>
<tr>
<td>Echo</td>
<td>The sales people do not generally have technical background and the engineering function is somewhat isolated from the environment. Although this has caused differences in knowledge stocks of these functions, there seems to be sufficient amount of common ground and frequent interaction.</td>
<td>Med</td>
<td>Product development is almost always behind schedules due to overstretching itself. However, Echo has been able to deliver products that can be characterized as best designed in the market with fast pace and sufficient amount of technical quality.</td>
</tr>
<tr>
<td>Golf</td>
<td>Golf has a clear separation of concerns between the four units that participate in product development decisions. Each function has its own area of expertise, and does not necessarily share much with other units. A moderate degree of conflict within the organization deepens the gaps between units. Sales staff largely consider the product as a black box.</td>
<td>High</td>
<td>Golf has a secure market position due to the large positive feedback effects in their business model. This, in combination of the relatively low-tech target market does not require high performance product development. It seems that the firm is not really innovative and relatively slow in product development, but can deliver sufficiently high quality.</td>
</tr>
</tbody>
</table>

*Table 2 Key variables*
<table>
<thead>
<tr>
<th>Firm</th>
<th>Description</th>
<th>Technological turbulence</th>
<th>Product complexity</th>
<th>Boundary spanning product owner</th>
<th>Speed of organizational change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>Most of the clients are in the public sector, which in general is not fast moving. Some new technologies have been introduced recently, but the product offerings have remained relatively stable over time. The product is an information system that is used with a web interface. High degree of understanding of customers processes is required to develop this system, but technically it is not particularly complex.</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Bravo</td>
<td>The product is new to the world, but the development of the underlying technology seems to be relatively predictable. Several Ph.D. level researchers are employed in the algorithm development. The software interfaces with third party information systems on customer servers.</td>
<td>Med</td>
<td>High</td>
<td>Med</td>
<td>Low</td>
</tr>
<tr>
<td>Charlie</td>
<td>The product is aimed on a market that is highly affected by fast development of communications and media standards. The product contains some of the latest media technology, but other than this, it is not particularly complex.</td>
<td>High</td>
<td>Med</td>
<td>High</td>
<td>Med</td>
</tr>
<tr>
<td>Delta</td>
<td>Technology enables new features that have not been possible earlier, but in sum the product technology has not changed much recently. The target market is relatively stable in terms of technological development. Large application software, which has complex data structures. Also, the application uses 3D graphics, which creates more complexity.</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Echo</td>
<td>Echo’s product relies on advances in communications technology. The market and different technical standards are still developing and the growth potential, but also the uncertainty is high. The software relies on third party servers and must run on several different platforms. The large number of configurations increases complexity of the relatively simple core product.</td>
<td>High</td>
<td>Med</td>
<td>Med</td>
<td>High</td>
</tr>
<tr>
<td>Golf</td>
<td>Golf has a secure market position with dominating market share. The target market is not very technology oriented, and hence things change only slowly. The product contains some sophisticated analysis functionality, but these do not seem to be deeply integrated to the product. The product is offered as ASP and there are only little connections to external systems.</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 3 Moderator and control variables
Analysis of our data from six cases indicates that there are often strong asymmetries of technical knowledge and market and use related knowledge. Both views are required, but if the gap between the engineering and marketing units is too wide, product development cannot proceed in an optimal manner. On the other hand, we found that too much consensus causes groupthink and reduces the understanding of the customers and the chance of identifying new trends in the markets.

In sum, the results are in line with existing research on cross functional integration, and provide evidence that the well known positive effect of heavy weight product champion has a positive effect in product development also in the context of software products. The main contribution of the paper is identification of the inverse U-shaped relationship between asymmetries of knowledge and product development performance. Somewhat surprisingly, we found that organizational change negatively moderated the negative effects of this main effect, but technological turbulence does not. One explanation for this is that technological turbulence affects product development performance directly. Finnish firms are often said to be technology focused, and hence it is not surprising that in this sample firms operating in technologically turbulent market seem to have better product development performance.

This study, like any other, is not without limitations. For making sure that the quality of our research is as good as possible, we have followed the generally accepted instructions by Yin (Yin and Campbell, 2002) for case studies in general and Eisenhardt (1989) and Miles and Huberman (1984) for data analysis. Reliability was enhanced through triangulation and by rigorously documenting the research process, and by using qualitative data analysis package to enable traceability of the coding process. The reliance on solely the first author’s judgement in coding can pose a threat to reliability, since due to confidentiality were.

Regarding sampling, we followed theoretical sampling using the polar opposites approach along two dimensions identified in the previous literature. While the exact ratings of the firms along these dimensions evolved during the coding, we achieved adequate spread. Using convenience sampling on the second stage of the sampling process should not pose threats to validity. However, a general weakness in all studies using a limited number of cases is that the generalizability can only be established in a theoretical, but not in a statistical sense. As a consequence, further research should focus on statistical testing of our theory. Another possible avenue for research would be to focus on the micro-processes of knowledge integration and the particular challenges create by the abstract nature of software as a product, perhaps through the means of an ethnographic study.

The practical implications of our results are twofold. First, we show evidence that not only the widely known gap of knowledge between the technical and non-technical personnel of software product firms is inevitable, but also that moderate levels of asymmetry are desirable. Our results emphasize the importance of boundary spanning product champion. However, contrary to earlier results, we show that this champion does not necessarily need to b the product manager, but any person with sufficient power can play the role.

References


