ICoRD '11
International Conference
on
Research into Design
Indian Institute of Science, Bangalore, 10 - 12 January, 2011.

Leveraging cross-functional knowledge for Product-Service System development: a knowledge life cycle perspective

Koteshwar Chirumalla, Andreas Larsson, Marco Bertoni, Tobias Larsson

Division of Functional Product Development, Luleå University of Technology, SE-971 87, Luleå, Sweden
{koteshwar.chirumalla, andreas.c.larsson, marco.bertoni, tobias.c.larsson}@ltu.se

Abstract: In recent years there has been a growing interest among product development organizations to capitalize on engineering knowledge as their core competitive advantage for innovation. Capturing, storing, retrieval and sharing of engineering knowledge from a wide range of enterprise memory systems has become crucial part of knowledge management practice among competitive organizations. Managing and reusing their knowledge can facilitate design engineers to make more timely and informed decisions, thus reducing the decision loops for new innovation projects. In light of a changing and dynamic enterprise definition, including a move towards Product-Service System (PSS) development, this paper discusses some of the limitations of current enterprise memory systems in reusing engineering knowledge across the proposed knowledge life cycle. Further, the paper illustrates how Web 2.0-based collaborative technologies can leverage cross-functional knowledge for new PSS development projects through an open, bottom-up, and collective sense-making approach to knowledge management.

Keywords: Cross-functional knowledge, Product-Service System development, Knowledge life cycle, Knowledge sharing, Engineering 2.0, Web 2.0.

1 Introduction

Engineering companies have traditionally focused their design and development activities on realizing physical artifacts (products). However, in today's competitive and global business environment, engineering companies are under pressure to bring product-service combinations into the market, which satisfies customer needs with minimum environmental impact. This emerging trend is commonly called Product-Service Systems (PSS) [1]. PSS development is a new business innovation strategy that combines design of both tangible products and intangible services in an integral way to fulfill increasingly sophisticated customer needs. The basic principle in a PSS paradigm is to offer the customer the function of a product and retain ownership and responsibility through all product life-cycle phases. Other related business trends that are commonly related to the

PSS concept include 'Functional Sales' [2], 'Functional Products (Total Care Products)' [3], and 'Service Engineering' [4]. For example, aircraft engine manufacturer Rolls-Royce has, at least partially, shifted its business model from the provision of physical engines to the provision of 'Power by the Hour' [5]. Earlier research studies [6-8] have shown that organizations would need to deal with several kinds of challenges to successfully offer PSS. In this paper, the authors are mainly focused on challenges concerning cross-functionality, and leveraging cross-functional knowledge in product life-cycle management.

1.1 Challenges of PSS development

Shifting towards a PSS paradigm places new demands and requirements on competence, capabilities and resources related to product and service design process. Hence, PSS development leads to a complex organizational structure with involvement of cross-functional teams from different stakeholders throughout the whole life-cycle, from understanding of customer needs to the development, use and disposal phase of the product. These dispersed, multi-disciplinary, cross-functional and cross-organizational teams, therefore, operate in a dynamic, multi-cultural, and highly unpredictable environment and create and manage enormous amounts of information and knowledge everyday in different form or different contexts by using a diverse set of enterprise software and system solutions [9]. The collaborative PSS development process requires enterprise-wide teams to consider the full range of product life-cycle needs in the earliest phases of the development project, especially when making trade-off decisions among many requirements and alternatives that impact the complete system. In most cases such decision rationale, if captured at all, is stored in corporate memory systems that are usually not widely available and accessible across corporate or even team boundaries, making it more difficult to retrieve the right information at the right time in the later stages of the product life-cycle. In PSS development, a major part of the value creation process is about developing a deep understanding of what customers need and value, an understanding which can be shaped from accessing a wide range of information, knowledge and communication related to customer activities in all life-cycle phases.

2 Literature Review

2.1 Cross-functional teams

Cross-functional teams (CFT) have emerged as a key coordination mechanism to work where no single individual or function possesses sufficient knowledge or skill for developing and managing innovative products and services. These teams typically include individuals from different functional areas within and outside of a company. Cross-functional teams can offer a variety of potential benefits to globally dispersed organizations in terms of productivity, creativity, decision-making, problem solving, and organizational learning [10]. For example, during the design process they can ensure that critical functional issues are addressed in the view of product life-cycle management and environmental sustainability. Although cross-functional teams have many advantages, earlier studies [11] identified cross-functional integration as one of the key challenges for successful business performance. However, virtual teams, spanning geographic and

temporal boundaries, do not normally have a shared history of working together, neither a shared knowledge base nor methods-systems-tools to create, store and share information and experiences [9]. Consequently in PSS development, it is fundamental to develop enterprise systems with fully integrated and social interactive features that support collaboration and sharing of both formal and informal knowledge to provide added value for an organization [12].

2.2 Knowledge life-cycle

Knowledge is classically defined as justified true belief, or true opinion combined with reason [13]. Knowledge has also been defined as information combined with experience, context, insight, interpretation and reflection [14]. One of the most common distinctions of knowledge in organizations has been identified as tacit knowledge and explicit knowledge [15]. Organizational knowledge is created through the process of knowledge conversion between tacit and explicit knowledge. Knowledge management (KM) is a set of practices that an organization applies to support this conversion process for acquiring, organizing and communicating both tacit and explicit knowledge [16]. Knowledge Life Cycle (KLC) frameworks are getting increasing, and several such frameworks have been proposed in the literature [17-20] with varying stages as shown in Table 1. Most of these frameworks are proposed with a view of developing a technology platform and enterprise system that incorporate life-cycle knowledge limited to a functional or organizational boundary. However, the requirements for the knowledge infrastructure become different in a PSS context, asking for methods and approaches better equipped to enable an open, bottom-up, collective sense-making approach to knowledge management [13, 21]. The authors propose a new framework for KLC (Table 1), that can support cross-functional teams across functional and geographical boundaries, and which is particularly focused on reusing knowledge from distributed cross-functional teams using a diverse set of enterprise systems.

Table 1 Knowledge Life Cycle frameworks

Authors	**Knowledge Life Cycle stages**
Blessing et.al., (2000)	Capture, Learn, Store, Retrieve, Use, Generate
Chan et.al., (2001)	Identify, Create, Transfer, Store, Reuse, Unlearn
Fruchter et.al., (2002)	Create, Capture, Index, Store, Retrieve, Reuse
Nuzzo et al., (2006)	Identify, Capture, Store, Access, Share, Use, Learn, Generate/Acquire
New framework: Chirumalla et al., (2010)	Capture, Store, Search & Locate, Retrieve, Share, Reuse

2.3 Engineering 2.0-A bottom-up and lightweight collaborative approach

The advent of social software (also called Web 2.0 [22]) has brought a new culture of sharing information on the Web where users can actively create, store, edit, access, share and distribute the content to larger audiences. Many organizations are starting to deploy Web 2.0 capabilities in their working environment to take advantage of corporate social networks and collective intelligence [23], and further to bridge the gap between social

and technical aspects. A recent 2009 McKinsey survey has outlined that more than 2/3 of 1700 companies worldwide have investigated or deployed Web 2.0 tools to support their product development activities [24]. Several initiatives are currently ongoing to integrate social features with traditional engineering tools. PTC, for instance, is exploring how to leverage social interaction and collaborative features, among global design teams, complementing CAD/PDM tools [25]. Some other examples are Dogear social bookmarking from IBM, Tagger from SolidWorks Labs, Vuuch and Windchill ProductPoint from PTC, and the Quest internal communications system from Microsoft. To capture the advent of social software adoption in product development, Larsson et al [7] coined the term Engineering 2.0 to indicate the use of Web 2.0 technologies in the fields of engineering and PSS development. Engineering 2.0 specifically targets globally dispersed engineering teams working in business-to-business situations of a Virtual Enterprise kind, where the available technology support for knowledge sharing still centers heavily on comparably "heavyweight" and top-down technologies like CAD, PDM, and PLM systems [7]. Engineering 2.0 is intended as a set of methods and tools to support bottom-up and lightweight knowledge sharing and stimulate the knowledge flow in social networks throughout PSS development. A set of these technologies, such as blogs, wikis, social networking, RSS feeds, tagging, micro-blogs, instant messaging, discussion forums and social bookmarking etc, and their application scenarios in PSS development have been explained in [12]. A categorization framework [26] for Engineering 2.0 tools has also been proposed to show to what extent the PSS development activity may benefit from the integration of different applications.

3 Methodological Approach

This study adopts a qualitative research methodology involving a case study approach [27]. The empirical findings are based on an industrial case study performed in manufacturing industry, in cross-functional product/service development projects. The case manufacturing company acts in a business-to-business context within the aerospace industry. The data has been collected through nine interviews with several cross-functional team participants at different hierarchical levels, which include engineers, managers, process owners, and project leaders, all with experience from global projects. Several forms of interview techniques, formal and informal discussions, and focus group meetings were used for the data collection. The data collected from the interviews and observations were recorded and transcribed, and key issues were drawn concerning the knowledge life-cycle, the essence of collaborative techniques, and supporting conditions. The issues were then analyzed with a PSS lens to figure out how existing enterprise systems could leverage cross-functional knowledge while moving towards PSS development.

3.1 Case study description

The case company develops and manufactures engine systems and components for the aerospace industry in co-operation with engine manufacturers. In the full supply chain towards the airliners, the company has design responsibility for components of aircraft engines. Also, they have one specific product where they are actually OEM (Original

Equipment Manufacturers) themselves and thus provide a lot more than a hardware product, i.e. providing all the functionalities around the engine like manual documentation, education, training, analysis tools, engineering support, maintenance, safety issues etc. As a first tier supplier, they have been increasingly involved earlier in design projects with the engine OEMs and aircraft OEMs to enable them to be better prepared from a technology perspective and also to prepare better for a long product life-cycle. For example, the company offers technical support to the engine manufacturers, which deliver the whole engine with a service responsibility period for 35 years. This virtual enterprise collaboration brings new demands and requirements on information management concerning different access and security levels with many more boundaries between organizations and inherent communication difficulties. Thus, it is important to define how a set of geographical dispersed companies can collaborate and manage their product information and organizational collaboration structure and how their respective workflows could operate together in the ideal virtual enterprise settings. This involves best practices, methods, tools and collaboration platforms, etc. In the case company, different people from different functions are involved in different phases of the product development process. Internally they mostly communicate through face-to-face meetings, informal meetings, phone and email conversations, and phone/web conferencing, etc. Externally, when it comes to collaboration between the companies, they are using web-based teamspaces with suppliers, and normally there is a lot of phone conferencing and email exchange with customers. It has been more difficult to collaborate with the customers since they normally do not set up shared teamspaces for these partnerships.

4 Empirical Findings

This section describes the empirical findings in each stage of the knowledge life cycle introduced above.

4.1 Knowledge Capture

The case company has the customer support personal at the customer sites, to assist answering questions that are outside the available documentation. All the requests from the customers in the product life-cycle are documented in a database; they usually document the background material for the change. Currently in the database there is just a lot of numbers in lists and it is often difficult to figure out how the decision was taken, what the common issues are and who makes the changes, etc. As one IT informant put it: *"I think we locked-in too much traditionally putting them in a document system like to die. Someone said that those traditional document systems are graveyards and I think they might have the point"*. On the other hand they also heard a lot about customer activities in their daily work, most of this information is stored in the email as unstructured information. In the design process, they have several iterations done by geographically distributed engineers before they get to a design to develop hardware from. The first few iterations are not likely to make it into the final design, but it is still useful to extract lessons learned that might arise in these early iterations. They usually capture the design rationale in design reports, excel sheets, design meeting notes, internal shared spaces, etc. Most importantly, it is often very difficult to find "why" explanations

for the decisions in databases. One of the design informants pointed out that, *"I think we should have some better tools for catching knowledge generally. But it is the big question how to form these tools more effectively"*.

4.2 Knowledge Store

The case company, in general, store all internal working documents and the supplier information in document management systems (DMS) and shared spaces with a lot of 'meta data', which allows them to have a more formal way of structuring the documentation. They usually store capabilities data, operation details, issue reports, problem reports, and project documents in the system. They have classified the document type in several classes like open, internal, and to OEMs, etc. Furthermore, they have described the rules related to how they should save them, what they should do with them, and how they should alter them, etc. However, for unstructured data types they lack document type and do not have any specific rules. In that case it is up to the person creating the document to decide whether it is open or internal. One of the project informants said, "*When I do a new project, how can I use this information that is stored today? And there are quite a lot of issues because they are stored in different systems and it is not presented in a way that I can use. If I get an issue, maybe I need to know what are the issues in previous project, how was the manufacturing process at that time, or materials that were used, what people were involved etc. I need to collect all the required data from different systems to understand the whole issue"*.

4.3 Knowledge Search/Locate

In the case company they cannot search a report from a previous project using free text search in the DMS, because they have to know which area it is about, and find out if there are any restrictions on accessing that area. Most of the time there is no certain template on how to do this search and acquire the information from databases. A design informant noted that, *"Our DMS is very bad in search but it's very solid, you archive things there and you almost never find them… maybe you cannot even see them, because you are in the wrong project. Maybe you need it or part of it for your own project. But you are restricted. You have to know what you are looking for and where to look, that is the major problem"*. One experienced informant pointed out that, *"We have a huge organizational memory of 80 years, but we do not have the possibility to search in it or retrieve it. We have to find a way to make it available and to get something from it"*. He further added, *"Today we do not have the time to search and retrieve the information. We want it like that. Instead of spending time on searching for, we create it once again, even if it is already somewhere in the cyber space. On the other hand, sometimes people forget to put 'meta data' with the documents, and then it's difficult to locate them in the databases"*.

4.4 Knowledge Retrieval

The information lies in separate databases and they have different levels related to access and security. If somebody does not have access, then they might not find the required information. If it says 'registered things' in the system, then only the persons working

within that area that can reach this information. Since they have many customers on several development programs, they cannot exchange/share the information in between them. Once they finish the project, they are not able to access the documents. One project informant described that, "*For example, if you do the lessons learned report and after a year you want to get access, even if you have written them by yourself you are not allowed because the project is closed. You maybe need some special access, we have a very rigid system to get access to, and it is unclear who has the access to do that"*.

4.5 Knowledge Share

As noted before, quite a lot of information today is restricted to a few people. As the company is not normally the OEM, they kind of depend on the tools their customers offer to collaborate externally. They have been working excessively on describing the business processes, standards and instructions in the existing tools, but these tools do not have that much functionality for commenting and other forms of relatively unstructured information. One design informant said, "*In previous project we were designing according to the customer's design system and they own the system, and we cannot share it with anybody else in the company. That's a problem getting documents from such a project because it is limited to the persons in that project"*. Another problem that informants highlighted was that, *"….information security part is a bit tricky, people don't really know what they can share or not…. Some persons say if they have access to more than one system, it's very supportive in their work, otherwise they just see their own information. That's a little difficulty when you think about knowledge sharing*". As they are 30 to 40 persons involved in the project, sharing information among team members and across different systems is a major concern. As one design engineer pointed out, *"If I want to share information between systems, how can I ensure what requirements I need to present to the provider?.... It is difficult to share the data within the company. If you want to be able to share the data outside to the different providers there are some more things to do"*.

4.6 Knowledge Reuse

Many engineering companies are having a hard time reusing the information from the existing enterprise systems. If they have rigid access rights, it is hard to find them even with a brilliant search engine. As one of the informants noted, "*We don't have a good integration between different databases/systems. So we have to look for all the documents/data in different systems from project name, materials numbers to identify the requirement numbers for those features you are interested in, and then you go into other systems and look for the capability data and the capability of the system*". A process owner informant highlighted that, *"I think we need to be stronger in taking care of lessons learned and make the lessons learned available for reuse in upcoming projects. From each project we should update project lesson learned documents once you reach a gate. But today we are not extremely good to take that information and actually feed it back to the system"*. All service knowledge from customer sites is stored in databases in the form of documents. This knowledge can be helpful in the earlier phases of other upcoming projects, but they do not have any regular activities for re-using this information and they do not have good practice at the company.

5 Discussion

PSS development places higher demands on the way engineers work and even require more dynamic processes to support concurrent cross-functional engineering teams to meet the customer demands with an effective business performances [4, 6, 9, 12]. As cross-functional teams have to realize life-cycle demands in the earliest phases of development projects, many of them have different views on how to design and develop a product in a long life-cycle view. Thus, they need more open, bottom-up and social shared platforms to support collective decision-making in an easy way. One of the experienced managers described that, *"I think the major difficulties during product development were that we are not using social networking tools as much as we can. We are not utilizing forums, blogs and wikis in great extent… …We need to get a level where we can support standard enterprise systems with wikis and blogs for co-ordinating and feeding informal information from the coffee room conversations and departmental /personal meetings. Then we bring the discussion into more open shared space where other persons can address relevant questions, give comments, and follow more open dialogues and moreover we can increase the network around certain area/issue"*.

The organization is trying to get everyone to use more social/bottom-up behavior around existing platforms like putting alerts, tags, RSS feeds and version handling things etc. There are even discussions about using Skype and Facebook for socializing in the work place. In their personal shared sites, everyone has the possibility to create a blog and a wiki if they like, and they have a shared document area where they can add documents and create temporary workspaces etc. Additionally, they also have a colleague tracker where they can add colleagues and follow their new documents and new blog posts.

The case company found benefits from using a blog as an open and shared platform across functional and organizational boundaries. One of the project informants said, *"The product development team needs to spread the information more in the projects, so respective designers know that first part went to workshop, manufacturing sequences and milestones etc…. It's more about getting design feedback and making sure that the design team is getting the feedback about the product in later stages of product development, etc."* They are trying to encourage design teams, supplier teams, product management and operators to post in the blog as well. Another informant added, "*Now if something happens during the manufacturing, the people can now go back and read about what happened at that time. We are also encouraging to put in the tags for easy searching and retrieval*".

Currently, the case company is developing a design practice system where there is an activity description of design process, how to do design iterations and criteria for selection, etc. One design informant pointed out that, *"… But this design practice system doesn't not really say the practical stuff. The wiki can make some distinction here; it can act more or less as an informal collector of different opinions and knowledge about practices. Therefore, having a wiki system connected to each activity, lesson learned and experiences are beneficial… we could document all design stages even better, so others can see and share their experiences*". Another project informant added: "*Often there are many lesson-learned documents in the company, but I think often it stops at the point of being documented and then nobody having time to review it. So the wiki is one way to leverage this knowledge"*.

Lightweight collaborative tools can facilitate to make the flow of questions and discussions more open and increasing the possibilities to get reflections from more competences across all the functional and organizational boundaries. For example, a wiki system could support more of a day-to-day process instead of waiting for a project gate. In that way it can support the gate passage and lessons learned process, to catch all potential lessons learned from different perspectives and then co-ordinate the information more towards possible process improvement in projects.

6 Conclusions and Future work

This paper emphasizes the importance of the creation of a collaborative environment where cross-functional teams can manage and reuse knowledge smoothly and informally across functions and corporate boundaries in PSS context. The empirical findings from case study support our argument concerning the limitations of current enterprise systems and propose how the cross-functional knowledge can be leveraged through the support of a more bottom-up and lightweight collaborative approach. Although several benefits were found with a bottom-up collaborative approach, several industrial obstacles were noticed in the fieldwork, such as: (1) establishing a culture that encourages working with unstructured information, (2) bridging the generation X and Y gap concerning adoption of social software, (3) making a cultural shift from sharing of information to sharing of knowledge, (4) allowing for more 2-way communications than 1-way communications, (5) motivating people to take active part in open dialogues.

A framework has been developed with regard to framing collaborative technologies in a knowledge life cycle perspective. Such a framework could provide improved decision support when reusing and leveraging cross-functional knowledge for innovation projects in the PSS domain. The collected needs and constraints from the case studies will enhance our understanding to move further in the development of a lightweight collaborative demonstrator tool in the proposed knowledge life cycle perspective.

Acknowledgments

We sincerely acknowledge the Faste Laboratory, a 10-year VINNOVA VINN Excellence Center focused on Functional Product innovation.

References

1 Baines, T.S, et al., "State-of-the-art in Product-Service Systems", Journal of Engineering Manufacture, 211, 1543-1552, 2007.

2 Lindahl, M. and G. Ölundh., "The Meaning of Functional Sales, Life Cycle Engineering: Challenges and Opportunities", 8th International Seminar on Life Cycle Engineering, CIRP, Varna, Bulgaria, 2001.

3 Alonso-Rasgado, T, et al., "The design of functional (total care) products", Journal of Engineering Design, 15, 6, 514-540, 2004.

4 Sakao, T, et al., "Service Engineering: A novel engineering discipline for producers to increase value combing service and product", Journal of Cleaner Production, 15, 590-604, 2007.

5 'Power by the Hour': Can paying only for performance redefine how products are sold and serviced?, Knowledge@Wharton, February 21, 2007.

6 Isaksson, O., Larsson, T.C. et al, "Development of Product-Service Systems: Challenges and opportunities for the manufacturing firm", Journal of Engineering Design, 20, 4, 329-348, 2009.

7 Mont, O., "Product-service systems: Panacea or myth", Ph.D. Thesis, The International Institute for Industrial Environmental Economics (IIIEE), Lund University, Sweden, 2004.

8 Bertoni, M. et al., "Coping with the knowledge Sharing barriers in Product Service Systems design, TMCE Conference, 2010.

9 Larsson, A., Ericson, Å. et al., "Engineering 2.0: Exploring Lightweight Technologies for the Virtual Enterprise", From CSCW to Web 2.0: European Developments in Collaborative Design, Springer, pp 173-191, 2010.

10 Parker, G.M., "Cross-Functional Teams: Working with Allies, Enemies & Other Strangers", Jossey-Bass Publishers, San Francisco, 1994.

11 Troy, L.C. et al., "Cross-functional integration and new product success: an empirical investigation of the findings", Journal of Marketing, 72, 6, 132-146, 2008.

12 Chirumalla, K., Bertoni, M. et al., "Exploring Lightweight Knowledge Sharing Technologies for Functional Product Development", 2nd CIRP IPS^2 Conference, 2010.

13 Hilpinen, R., "Knowing that One Knows and the Classical Definition of Knowledge", Synthese, 21, 2, 109-132, June 1970.

14 Davenport, T., Prusak, L., "Working Knowledge", Harvard Business School Press, 1998.

15 Nonaka, I., "The knowledge-creating company", Harvard Business Review, 69, 96-104.

16 Alavi, M., D. Leidner., "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," MIS Quarterly, 25,1, 107-136, March 2001.

17 Blessing, L., et al., "Supporting the Knowledge Life-Cycle", Knowledge Intensive ComputerAided Design, Finger, S., Tomiyama, T. And Manyla, M. (Eds.), Kluwer Academic Publishers, 2000, pp.21-38.

18 Chan, R., Rosemann, M., "Managing Knowledge in Enterprise Systems", Journal of Systems & Information Technology, 5, 2, 37-54, 2001.

19 Fruchter, R., Demian, P., "CoMem: Designing an interaction experience for reuse of rich contextual knowledge from a corporate memory", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 16, 127–147, 2002.

20 Nuzzo, P., Lockwood, F., "D3.1.2_2 - Knowledge Enabled Solution Components State of the Art", VIVACE project official deliverable, 2006.

21 Bertoni, M., Larsson, T., "Requirements for a KEE systems supporting PSS development", International Conference on Research into Design, ICoRD'09, 2009.

22 O'Reilly, T., "What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software", 2005.

23 McAfee, A., "Enterprise 2.0: The Dawn of Emergent Collaboration", MIT Sloan Management Review, 47, 3, 21-28, 2006.

24 Bughin, J., "How Companies are benefiting from Web 2.0", McKinsey Global Survey results, June 2009.

25 Shoemaker, T., "Social Networking and Product Development", Cadalyst, 23 July 2009.

26 Bertoni, M., "Bottom-up Knowledge Sharing in PSS design: A classification framework, 11th Design Conference, Dubrovnik, 17-20 May 2010.

27 Yin, R.K., "Case Study Research: Design and Methods", Sage Publications, 1994.