Exploring Communities of Practice for Product Family Engineering

Tor Erlend Fægri[1], Torgeir Dingsøyr[1], Letizia Jaccheri[2],
Patricia Lago[3], Hans van Vliet[3]

[1]SINTEF Information and Communication Technology
NO-7465 Trondheim, Norway
[2]Dept. of Computer and Information Science,
Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
[3]Vrije Universiteit, Amsterdam, The Netherlands

Abstract. Product Family Engineering is an approach to software engineering that seeks to reduce the global effort in producing multiple software products by actively governing the reuse of assets between the family members. However, Product Family Engineering is a highly knowledge intensive activity putting stringent demands careful planning and management in the organization. This paper presents a study proposal that will examine the effects of knowledge brokering among Communities of Practice as a means to assist this activity.

1. Introduction

Software engineering is a knowledge intensive activity. The demand for increasingly complex and sophisticated products fuels the search for more efficient approaches to software engineering. Modularization and component-based software engineering techniques have been widely adopted by organizations producing software products of some size and allows for significant flexibility in the development process. Modules can be developed in parallel or may be supplied by external organizations. Simultaneously, increasing complexity has lead to extensive degrees of specialization among the workers.

Many organizations want to exploit the potential in component-based software engineering to produce multiple products based on common assets. We call this Product Family Engineering (PFE). PFE seeks to optimize the resource usage in this production process through management of business, architecture, process and organization [1]. That is, through the appropriate control of these aspects to maintain a set of assets that enables the least resource demanding production of a set of products.

It is clear that PFE adds another level of complexity to the software engineering process. A result of this is that the individuals are faced with the challenge of managing profound and deep knowledge related to performing their own tasks. Specialization increases the boundaries between the individuals, hindering creative collaboration among the groups and may ultimately lead to reduction in job satisfaction. As little research is done in the field of knowledge management in PFE organizations so far, we take a rather open approach in our investigations.

Our research questions are:

1) Are there particular aspects of PFE that can benefit from structured knowledge management practices?
2) Can brokering activities from Communities of Practice support PFE organizations in ensuring good knowledge distribution among the workers?

2. Product Family Engineering

As the complexity and size of software systems increases people needs methods, processes and principles to cope with the scale. One commonly used principle is *divide and conquer*. This may manifest itself as recursive modularization and component-based development. That is, at a certain level of complexity, the software system is partitioned into multiple levels of modules or components (i.e. modules consisting of sub modules, components consisting of sub components). As the complexity reaches a certain level, these components are likely to be developed by different people, different parts of the organization or even different organizations. After some time, many organizations will establish a repository of such assets, including significant experience related to their use and limitations.

Now, the company might see a potential to exploit this repository to support a product family – for example with the objective to target different markets with somewhat similar products, using distinct combinations of these assets. This reduces the amount of product specific code that goes into each product, thus promising to reduce cost of development. Similarly, valuable experience related to the use of the assets is maintained. We believe it is important to understand aspects related to how the organization can use this experience to support continuous learning. The figure below illustrates this goal.

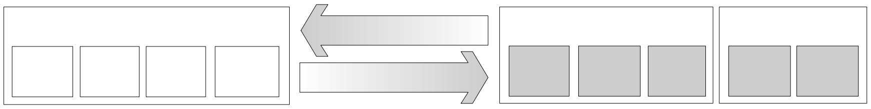

Fig. 1. Product derivation and asset reintegration

Achieving these ambitions requires elaborate planning, operative support and careful attention to organizational capabilities [2]. An often touted benefit of PFE is the potential for streamlining the product production process by organizing the workforce into two groups, one building reusable assets, and one building products from these assets. In this way, the organization can foster specialist knowledge among the people developing reusable assets while allowing people with solution building strengths to exploit the asset repository in assembling complex products. However, the effects of this segmentation have not yet been thoroughly investigated [1]. We assume that knowledge management practices play a key role in gaining these benefits, but we want to investigate this further.

Capturing, representing and sharing domain knowledge in a PFE organization can be assisted by artifacts such as patterns, frameworks and reference architectures [3]. Also, proper knowledge representation eases communication of architectural principles through the development organization [4]. However, the organizational challenge of ensuring efficient use of these, and other tacit knowledge bearers have been given little attention from the scientific society. Additionally, relevant domain knowledge not only concerns the structure and contents of the domain, but also organizational and managerial aspects. For example, if a company has a divisionalized structure such as a set of business units, this may impact its software assets. Capturing and explicitly documenting such attributes enriches our knowledge of these assets. Furthermore, experience with PFE evolution shows that it is often hampered by tacit assumptions. These may be technical (such as standards concerning the underlying infrastructure on which products rely), but equally often they are managerial or organizational in nature [5]. Thus, we believe it is worthwhile to investigate knowledge management practices within Product Family Engineering organizations further.

3. Communities of Practice and brokering

Communities of Practice can be defined as ”Groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis” [6]. Such communities differ from normal business units in that their purpose is knowledge-creation, and they are usually loosely connected, self-managed and informal. Communities of Practice enable the capturing and transfers of tacit knowledge by letting people from different departments in an organization discuss common topics of interest.

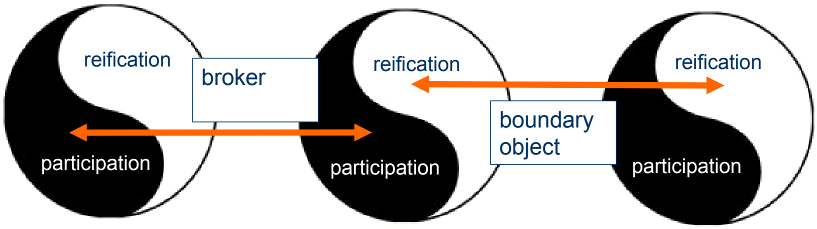

Fig. 2. Communities of Practice

Communities of Practice are said to consist of two main parts: *participation* in the community – interacting and “active involvement in social enterprises” [7] and *reification*, the process of creating artifacts from the community, like documents (see figure at right).

Knowledge sharing between Communities of Practice is referred to as *boundary relations,* where connecting communities through participation is called *brokering,* and through reification *boundary objects.* Many practices can classify as brokering – like job rotation, physical design of work-environment to stimulate cross-sectional discussions. Typical boundary objects are documents that are shared between communities such as roadmaps and design documents. Organizations report that such communities help improve communication and save time by “working smarter” [6].

The community can be organized around regular meetings, face to face or on-line, but also interaction through mailing list or news group discussions between meetings. Communities often also include a role as a contact point to get relevant knowledge in a topic area for people not participating in the community. Because most people who are involved in a Community of Practice have their main responsibility in one of the business units, it is important to organize it in a flexible manner. Work should not depend on particular individuals being active at all times. Usually, one person in the core group is appointed as coordinator with the responsibility of arranging meetings.

4. Suggested study

As depicted in Fig. 3 (below), PFE represents knowledge about the product families mainly from two perspectives: the representation of the common and variable features in the product family (feature models - depicted in the left part of the figure) and the representation of technical assets like code components (structural models - depicted in the right part of the figure). Feature and structural models can be used by the Communities of Practice as the core mechanisms for brokering: feature models are domain-level representations that can be easily understood by people that want to decide on the functionality of products; structural models are representations (of generic/reusable and concrete/product-specific solutions) that can be used by people producing or using reusable assets.

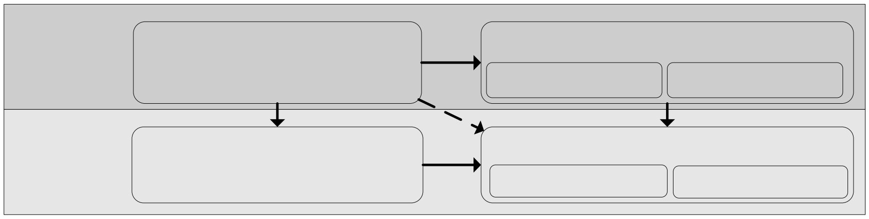

Fig. 3. Product Family modeling framework

Further, PFE knowledge can be represented at the family level (in the top half of the figure above) and at the product level (in the lower half of the figure). The Product Family level describes the aspects shared by all products in the family, whereas the Product level focuses on aspects that are specific to individual products.

The arrows represent traceability links that model the derivation dependencies between models. They can be navigated to share knowledge about artifacts. For example, the horizontal arrows (between feature model and structural model) can be used to learn how a feature has been implemented by a selected component. The vertical arrows (crossing the two levels) can be used to learn if/how elements at the product family level are supported by corresponding elements in different products (e.g. which features shared by the product family are supported in which products, or how a product family component is refined within a selected product).

For a product family to be successful, it is important to understand the factors related to adoption and refinement of reusable assets in the products. It is also important to understand how these aspects relate to the scoping of the PF level feature models, i.e. the ability of the asset repository to reflect the real needs in the product development groups. This is paramount in order to maintain a vital repository that is able to support the required variability in the product family. It is an enabler for organizational learning.

We intend to investigate how knowledge sharing through brokering between people working with reusable assets and products affects the dynamics of the product family, in particular the level of alignment of the repository of reusable assets at the system family level compared to the assets actually used in the delivered products. We will also study the effects of knowledge brokering with regards to reintegration of capabilities developed in the product organizations back into the system family.

Due to lack of validated metrics [8, 9], we have chosen a qualitative study. Through semi-structured interviews, we suggest studying two different organizations working with product families, and choose a) a company that has few or no brokering between reusable asset and product departments, and b) a company that has many brokering practices in place. We will study brokering on functional and non-functional aspects of assets, as well as the extent to which organizational and managerial aspects that impact the assets, are taken into account, or had better be taken into account.

References

1. Linden, F.v.d., *Software product families in Europe: The ESAPS and CAFÉ projects.* IEEE Software, 2002. **19**(4): p. 41-49.
2. Bühle, S., et al. *Exploring the context of product line adoption*. in *5th International workshop on Product Family Engineering*. 2003. Siena, Italy: Springer Verlag.
3. Hallsteinsen, S., T.E. Fægri, and M. Syrstad. *Patterns in Product Family Architecture Design*. in *5th International workshop on Product Family Engineering*. 2003. Siena, Italy: Springer Verlag.
4. Mustapic, G. *Real World Influences on Software Architecture - Interviews with Industrial System Experts*. in *WICSA-4*. 2004. Oslo, Norway.
5. Lago, P. and H.v. Vliet. *Observations from the Recovery of a Software Product Family*. in *Third software product line conference*. 2004: Springer.
6. Wenger, E.C., R. McDermott, and W.M. Snyder, *Cultivating Communities of Practice*. 2002, Boston: Hardvard Business School Press.
7. Wenger, E.C., *Communities of Practice: Learning, meaning and identity*. 1998: Cambridge University Press.
8. Fenton, N. and M. Neil, *Software metrics: A roadmap*, in *The future of software engineering*, A. Finkenstein, Editor. 2000, ACM Press: New York.
9. Seaman, C.B., *Qualitative Methods in Empirical Studies of Software Engineering.* IEEE Transactions on Software Engineering, 1999. **25**(4): p. 557-572.