

A Collaborative Approach for Reengineering-based Product Line Scoping

Muhammad A. Noor
*Institute for Systems
Engineering & Automation
Johannes Kepler University
4040 Linz, Austria
man@sea.uni-linz.ac.at*

Rick Rabiser
*Christian Doppler Laboratory
for Automated Software Engineering
Johannes Kepler University
4040 Linz, Austria
rabiser@ase.jku.at*

Paul Grünbacher
*Christian Doppler Laboratory
for Automated Software Engineering
Johannes Kepler University
4040 Linz, Austria
paul.gruenbacher@jku.at*

Abstract

Product line scoping is an important activity in re-engineering-based software product line adoption. Both business issues and technical concerns have to be handled adequately. However, involving stakeholders representing these different concerns is not straightforward. Agile methods strongly emphasize stakeholder involvement and customer collaboration. In this paper we propose a collaborative approach which is intended to complement existing approaches for re-engineering of legacy products in product line engineering. It supports success-critical stakeholders working collaboratively to converge on a product map and a definition of reusable infrastructures. Our approach is based on the WinWin requirements negotiation method which facilitates stakeholder collaboration and guidance towards mutually acceptable solutions.

1. Introduction

An increasing number of organizations are shifting to a software product line (PL) approach to reduce time to market and development costs while at the same time increase reliability and ease maintenance [9]. Often product lines are not developed from scratch. Rather they are introduced to already existing products [12][17]. Research on reengineering-based PL introduction is typically focused on the more technical aspects (e.g., [1]). Stakeholder collaboration has received less attention so far.

Reengineering processes in PLE are most effective when business objectives are reconciled with technical constraints. A critical task when developing a product line is scoping. Comprehensive scoping approaches exist for the development of new product lines (e.g., [18]). However, comparably limited methods and

guidelines are available for scoping of product lines which are to be introduced to existing products. However, economic pressures necessitate the reengineering of legacy assets. Activities to assess the domain potential and scope the reuse infrastructure are considerably different and more complex for a reengineering-based approach than for developing a new PL from scratch.

Reengineering based PL adoption is an internal organizational task (as there is no external client); which is more suitable to be conducted in an incremental fashion in order to manage risks and optimize the process continuously. However, there is a strong need for team members to have a common vision about the overall project. We suggest that this can be achieved by performing the PL scoping upfront in a collaborative way.

In the PuLSE framework [2] as later described in [18] the aims of scoping are to develop a product map and to identify and assess reusable assets which will provide maximum business profit. A product map is a simple but powerful tool which shows the overall PL in concise manner. Scoping in this context requires involvement of management, marketing, and sales personnel in order to address market and business aspects. No single stakeholder has all information necessary to estimate the cost of incorporating legacy assets into core assets for future reuse. In order to maximize the benefits for a given limited budget it is necessary to have a good estimate of cost and benefits. Benefits can be estimated by projecting historical sales data. Cost can be estimated with the help of developers and architects. At the same time stakeholder goals must be congruent with technical constraints of existing products. Architects, developers, and maintenance staff have the technical knowledge about the existing system. This knowledge is vital for scoping a PL for existing products but is distributed among different stakeholders. Mutually agreed solutions can therefore only be

achieved through close collaboration and knowledge sharing.

In this paper we thus motivate the need for a collaborative scoping approach which focuses on early involvement of stakeholders representing business and technical domains. We present an initial collaborative, stakeholder-centric approach for PL scoping, which can facilitate the incremental migration process from a conventional to a PL approach. Our approach is also consistent with agile principles, of collaboration, shared vision, and incremental development. We do not focus on bottom-up approaches which deal more with the technical aspects of reengineering, like for example in [15].

Our approach is grounded in the well-documented and proven approaches PuLSE [2] and EasyWinWin [5][7]. Our aim is to combine and tailor these approaches to the described context. We use the three Pulse activities ‘product mapping’, ‘domain potential analysis’, and ‘reuse infrastructure scoping’ as a general framework [18]. EasyWinWin is a collaborative groupware-supported requirement negotiation method based on Boehm’s theory-W [6]. It is supported by several tools; however in this paper we adopt a Group Support System (GSS) to demonstrate our ideas. In [13] it has been shown that EasyWinWin complements and supports agile methods by supporting the attainment of a shared vision, the identification of critical stakeholders and their extensive involvement. .

The remainder of this paper is structured as follows: In Section 2 we discuss success-critical stakeholders and highlight why their participation in product line scoping is essential. We present our approach in Section 3. There we also briefly discuss how collaboration patterns helped us in defining the collaborative process. In Section 4 we present the GSS tools and show a prototype. Section 5 concludes this paper and presents future research plans.

2. Typical Stakeholders in PL Scoping

Agile methods emphasize the involvement of people in software engineering. The identification of stakeholders and their involvement is also central to our proposed approach. Unresolved conflicts between stakeholders can jeopardise a project’s success. A collaborative negotiation approach in PL scoping can help to identify and overcome such conflicts to attain mutual agreements and a shared vision among stakeholders. Following Boehm’s spider web [4] we describe the success-critical stakeholders in the PL scoping context and briefly summarize their typical concerns:

Senior management: The decision to adopt a product line approach often comes from senior management which outlines business objectives and allocates time and resources for the project. Senior management is typically interested in cost-effective, timely, and robust solutions which deliver the desired results with minimal exposure to risk. After the commencement of the project senior management has to ensure that business focus is not lost due to technical concerns.

Customers: In most cases it is not feasible to directly involve customers in internal product reengineering processes. Customer concerns and expectations have to be ascertained and incorporated through representatives. For example, marketing and sales people may bring in their knowledge about the market and detailed needs of different customers.

Marketing and sales: Marketing and sales departments have to generate revenue for the organization. They rely on products which are desirable for customers and meet their needs. Marketing and sales people are essential to represent the business concerns in a PL adoption project. They have full access and understanding of historic sales data and feedback from customers. Based on that, future sale projections are made which help estimating the quantitative benefits of the project.

Software architects: Architects know the legacy products which are to be reengineered. There are many inherent complexities in reengineering-based product line development, e.g., when independent individual legacy products are to be merged in a single platform (PL infrastructure). There may be significant differences in terms of product architecture and implementation technology along with numerous combinations of differences in non-functional attributes. The aim of reengineering is not only to enable reuse during development of core assets. Core assets should be developed in such a way that unique features of different products can also be easily reused. This makes early involvement of architects essential.

Developers: The participation of developers is essential to assess different alternatives of legacy assets from a reusability perspective. Their knowledge facilitates estimating the costs of changing different aspects of legacy products.

Maintenance personnel: Similar to developers, maintenance staff is highly valuable in the early stages of product line scoping. Their intimate knowledge of the strengths and weaknesses of existing products is essential. Feedback from the maintenance staff enables the evaluation of different PL evolution options to ease transition to customers and to ensure product compatibility.

3. Collaborative Approach to PL Scoping

The benefits of involving different stakeholders in PL scoping are fairly obvious. The knowledge required for PL scoping is distributed among different people in the organization. A collaborative approach can facilitate understanding the different stakeholder concerns and converging on mutually acceptable solutions.

Our collaborative reengineering-based product line scoping approach uses PuLSE [2] as a general framework. For defining a collaborative process we use guidelines and patterns from collaboration engineering [8] to improve interaction and decision-making of stakeholders.

Achieving effective team collaboration is a considerable challenge [10]. The area of collaboration engineering [8] aims to define repeatable work practices for recurring team tasks. These practices are based on the assumption that team processes are much more effective when explicitly modeled and structured.

Collaboration engineering identifies five general patterns of collaboration that serve as building blocks for the conceptual design phase of a collaborative work practice. Each pattern characterizes a way in which a group can move towards achieving its goal. The patterns are *diverge*, *converge*, *organize*, *evaluate*, and *build consensus* [8]: In *diverge* a group moves from having fewer to having more concepts with which to work (e.g., a brainstorming activity). In *converge* people move from having many to a focus on a few concepts worth more attention, and move from less to more shared understanding of concepts and labels (e.g., a moderated discussion about the brainstorming results). *Organize* means that a group starts to understand the relationships among concepts (e.g., structuring brainstorming results into different categories). *Evaluate* aims to understand the instrumentality of concepts (e.g., voting to understand the preferences of stakeholders). *Align goals* makes more individuals willing to commit to a proposal (e.g., discussing situations where a team does not agree). Collaboration patterns are typically supported with collaborative tools such as electronic brainstorming, group categorization, group outlining, or voting tools, to name but a few.

EasyWinWin [7] is an example of a collaborative process that makes use of these patterns of group interaction. This requirements negotiation method is based on Boehm's WinWin approach and complements the high-level WinWin approach with more detailed negotiation guidelines and tool support [14][13][5][7]. The potential benefits of adapting EasyWinWin in our reengineering-based PL scoping scenario are manifold: (1) The method is a well-documented and proven in real-world settings, (2) it is an easy to use by practitio-

ners, and (3) facilitates group collaboration through guidelines and proper tools.

3.1. Key steps of the scoping process

Product mapping, domain potential analysis, and reuse infrastructure scoping are key activities for reengineering-based product line scoping [2][18]. When preparing a product map the products which will be part of the product line are selected and features and domains are determined. During domain potential analysis the size and suitability of a domain for serving as a core asset are assessed. Furthermore, collateral costs of interdependencies and conflicts among features of the domain and rest of the product(s) are identified. In reuse infrastructure scoping the potential costs and benefits to develop a domain as core asset are quantitatively determined. Domains which promise to yield the highest return on investment are identified and scoped unambiguously.

Collaboration is essential in these three scoping activities:

Product Mapping: A product map contains domains, features, and products in tabular form and helps to visualize commonalities and variabilities of a PL at feature level. Product mapping starts with domain identification. Domains which exist in legacy products and domains planned for the future constitute a product map. We can assume that stakeholders are well versed with the products and the products' domains. However, it is probable that not all stakeholders agree upon the domain definition. Disagreements about features and/or domains need to be revealed and discussed in order to achieve a common understanding and to attain consensus. While scoping domains through feature assignment, features can be discussed, refined, and consolidated. Products to be in the PL also need to be discussed because it is possible to develop new products by using existing features in a different combination. Every existing product that is not part of the PL and new feature combinations should also be considered. Therefore strong input from business people is required. Stakeholders have to agree on the assignment of features to products. As noted earlier business needs should be compatible with technical realities. Therefore features have to be prioritized according to their business value and the technical feasibility, from a reengineering perspective. Again possible disagreements need to be revealed and discussed to build consensus.

Domain Potential Analysis: In domain potential analysis domains are identified which have the highest potential for reuse and promise the greatest return on investment for reengineering them as core asset for the

product line. This activity is less challenging in the reengineering context because products are already successful in the market and the organization already has experience with the domain. Each domain is analyzed for certain parameters such as the size of the domain, reuse frequency in the future, complexity of reengineering, cost of reengineering, or cost of developing it from scratch. In our context it is important to mutually agree on the domains which will be probed quantitatively later and exclude domains which are low value candidates.

Reuse Infrastructure Scoping: The outcome of a domain analysis can be transformed into quantitative benefit measures. However, it is not feasible to tackle this technical problem at the stage of negotiation. This task can be best performed in tandem with reverse engineering (e.g., Pulse DSSA [1]) or after that. In our case it is important to not only scope the selected reusable assets at product map level but to also identify assets at component (infrastructure) level. This is not trivial because in many cases the traceability between features and the technical-solution's components is not readily visible [15]. In such a situation preliminary identification of the components belonging to selected domains is a useful input for the reverse engineering team. It can help them to visualize the context and to focus their efforts on more relevant areas.

3.2. Defining the collaborative process

In this section we show how the discussed tasks and activities can be fit into a collaborative process based on patterns and guidelines from collaboration engineering. In order to come up with the collaborative scoping process, we analyzed generic activities which are performed in traditional scoping process. We grouped and tailored those activities, according to the specific needs of the context under discussion. Table 1 shows the activities of the process together with their purpose and collaboration pattern. The collaboration patterns are technology-neutral and do not assume a certain collaborative tool. In fact one could also use paper & pencil-based techniques to enact the collaboration patterns in Table 1.

The process follows the general EasyWinWin negotiation process by defining the scope of the meeting (1), gathering input from the team and organizing the collected ideas (2, 3), getting group input of preferences (5, 6) and revealing and discussing situations where the group members do not agree (7). The results are mutually agreed domains, features, and products, scoped for reuse, based on business value, and technical feasibility for reengineering.

Table 1. Collaborative, reengineering-based scoping process

Activity	Purpose	Collaboration Pattern
(1) Review and revise domains	Identify and agree on domains contained in the legacy products	Converge
(2) Brainstorm features for domains	Collect the features for different domains from stakeholders. Use one brainstorming session per identified domain	Diverge
(3) Converge on features per domain	Discuss the brainstormed features, eliminate redundant information through moderated discussion, find a common level of feature granularity in the team	Converge Organize
(4) Agree on the products in the PL	Elicit and agree on the products which will be part of the PL, existing and planned for the future, in order to achieve common understanding	Diverge Converge
(5) Assign features to products (vote)	Participants assign features to different products. Reveal differences & revote if necessary	Evaluate Align goals
(6) Prioritize & optimize features for business value & feasibility	Prioritize features according to business value using input of prior market studies and input from relevant stakeholders. Determine the technical feasibility of features from reengineering context	Evaluate
(7) Reveal differences & revote	Possible differences are revealed and reasons are discussed to achieve consensus. A revote is initiated after the discussion. Decide on feature evolution (drop, continue, modify)	Align goals

The process is currently focusing on product mapping. As discussed above this activity is essential in reengineering-based product line scoping. However, we will include the other two activities in the process after receiving further feedback from feasibility studies. In the beginning of the process, possible domains are discussed. During this activity the benefits and costs of domains can be explored thereby considering domain potential analysis.

After step 3 (converge on features & domains) the scoping of domains is completed. Each domain can then be assessed for its suitability for reuse as infrastructure and the benefits of doing so can quantitatively be assessed. Thereby partly the activity of reuse infrastructure scoping is also covered and the other half of this activity (reusable infrastructure scoping) should be timed together with reverse engineering activity as noted earlier.

4. Providing tool support with a GSS

Group Support Systems (GSS) facilitates and trigger interaction in team. The allow to support groups without controlling the process in a rigid or prescriptive way. GroupSystems is a commercially available Group System Support (GSS). It supports the modeling

of collaborative processes and provides capabilities for collaborative tasks. The success of a GSS for collaborative processes is well documented for software engineering tasks such as requirements definitions [7] and software inspection [3]. Selected tools relevant for our purpose provided include:

Group Outliner. This tool enables participants to create a tree-based outline. The tree provides a structure for collecting ideas. During discussions the tree can easily be modified. The tool can for example be used to collaboratively work on a feature model.

Brainstorming. Using this tool allows a group to rapidly and effectively collect ideas. Experiences show that a team of 10 people can collect 300+ ideas in 45 minutes.

Categorizer. This tool is used to organize ideas into different categories. Categories and the content of categories can easily be modified and moved to other categories (see Figure 1).

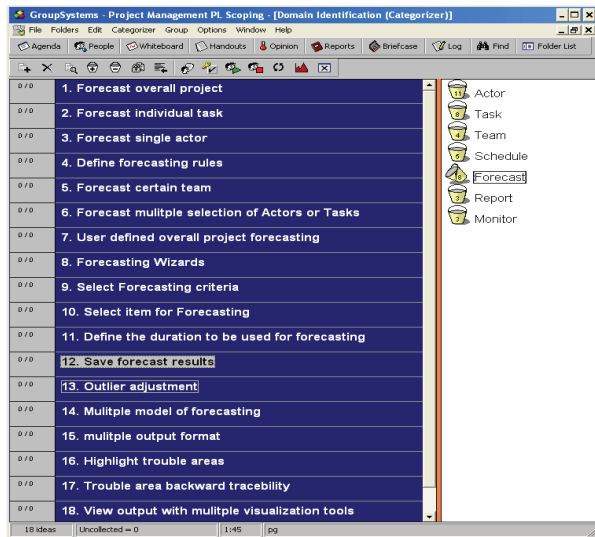


Figure 1. Converge on features & domains (GSS Categorizer)

Alternate analysis. This tool provides voting capabilities allowing a team to evaluate different alternatives against an arbitrary number of criteria. The tool also has powerful features to reveal situations in which a team of stakeholder can not agree on an issue. This capability supports building consensus in a team.

We conducted an internal prototype study using the tools for a preliminary evaluation of our initial process. Our case study is the open source project management tool Ganttproject for which we started the definition of a product line. We performed the scoping exercise as outlined in this paper. We wanted to find out whether knowledge required at different steps of our process can be presented in a structured manner using GSS and

its constituent tools. Figure 1 shows a snapshot of the activity ‘Converge on features & domains’. During this activity features collected in the brainstorming session are collaboratively assigned to different domains using the categorizer tool.

The categorizer is used by the moderator of the workshop on a public screen. The participants view the collected features in the brainstorming tool. The group converges on a joint list of domains and features in a moderated discussion that takes the brainstorming results as input.

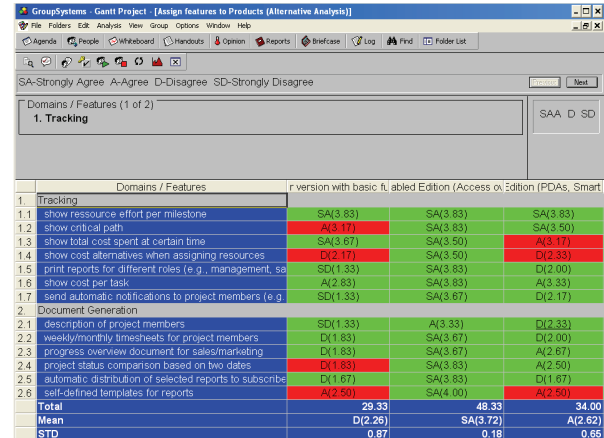


Figure 2. Voting result on a product map.

The GSS voting tool is convenient for quickly gathering participants’ opinions and for building consensus. Figure 2 shows the alternative analysis voting tool with voting results on a product map (i.e., activity ‘assign features to products’). Each participant gets an electronic ballot to enter his preferred mapping. Later on the tools highlights situations where people could not agree using red cells. By considering the reasons for disagreement stakeholders reveal hidden assumptions and share knowledge in the team.

5. Conclusions and Further Work

Product line scoping is an important activity with serious economical and technical implications. Scoping is necessary for both developing a PL from scratch and for reengineering-based PL adoption. However, in the latter case the focus lies more on the technical consideration of the assimilation of different products, domains, and features into one integrated infrastructure. Due to the distributed nature of the knowledge necessary for scoping, a collaborative approach is required.

There exist few consolidated approaches which can guide the scoping process. We presented in this paper such an approach which combines good practices of

PulSE, EasyWinWin, and collaboration engineering. The approach facilitates stakeholder involvement collaboration, a practice advocated by agile methods. Preliminary experience indicates that our approach is useful in supporting reengineering scoping process.

The need for our work is confirmed by reports on lessons learned when introducing product lines in industry. For example, Ebert *et al.* [12] report that a clear business focus, strong release planning, and requirements management are essential for the success of such an exercise. In [11] Dhungana *et al.* underscore the significance of sharing architectural knowledge among different stakeholders in PLE.

In the future we will conduct a complete case study to fully explain and validate the process in our ongoing collaboration with a large industrial organization which is currently introducing a product line.

References

- [1] J. Bayer *et al.*, "Definition of Reference Architectures Based on Existing Systems", *IESE-Report 034.04/E*, 2004.
- [2] J. Bayer *et al.*, "PuLSE: A Methodology to Develop Software Product Lines", *5th Symposium on Software Reusability (SSR '99)*, 1999.
- [3] S. Biffli, P. Grünbacher, and M. Halling, "A Family of Experiments to Investigate the Effects of Groupware for Software Inspection", in *Journal on Automated Software Engineering*, Springer, 2006 (to appear).
- [4] B.W. Boehm, D. Port, and M. Al-Said, "Avoiding the Software Model-Clash Spiderweb", *IEEE Computer*, 33(11), 2000, pp. 120-122.
- [5] B. Boehm, P. Grünbacher, and R.O. Briggs, "Developing Groupware for Requirements Negotiation: Lessons Learned", *IEEE Software*, 18(3), 2001, pp. 46-55.
- [6] B.W. Boehm and R. Ross, "Theory-W Software Project Management: Principles and Examples", *IEEE Transactions on Software Engineering*, 15(7), 1989, pp. 902-916.
- [7] R.O. Briggs and P. Grünbacher, "EasyWinWin: Managing Complexity in Requirements Negotiation with GSS", *35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, Big Island, Hawaii, IEEE CS, 2002.
- [8] R.O. Briggs, G.J. de Vreede, and J.F. Nunamaker, "Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems", *Journal of Management Information Systems*, 19(4), 2003, pp. 31-63.
- [9] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*, SEI Series in Software Engineering. Addison-Wesley, August 2001.
- [10] G.J. de Vreede and R.O. Briggs, "Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks", in the *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, 2005.
- [11] D. Dhungana, R. Rabiser, P. Grünbacher, H. Prähofer, C. Federspiel, and K. Lehner, "Architectural Knowledge in Product Line Engineering: An Industrial Case Study", to appear in *Proceedings of the EUROMICRO 2006 Conference on Software Engineering and advanced applications (SEAA)*, Cavtat/Dubrovnik, Croatia, 2006.
- [12] C. Ebert and M. Smouts, "Tricks and Traps of Initiating a Product Line Concept in Existing Products", in the *Proceedings of the 25th International Conference on Software Engineering (ICSE'03)*, 2003, pp. 520-527.
- [13] P. Grünbacher, C. Hofer, "Complementing XP with Requirements Negotiation", 3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering, Alghero, Sardinia, Italy, 2002.
- [14] P. Grünbacher, M. Halling, S. Biffli, H. Kitapci, and B.W. Boehm, "Integrating Collaborative Processes and Quality Assurance Techniques: Experiences from Requirements Negotiation", *Journal of Management Information Systems*, 4(20), M.E. Sharpe Inc., 2004, pp 9-29.
- [15] P. Grünbacher, A. Egyed, and N. Medvidovic, "Reconciling Software Requirements and Architectures with Intermediate Models", *Journal for Software and System Modeling (SoSyM)*, 3(3), August 2004, pp. 235-253.
- [16] J. Knodel and D. Muthig, "Analyzing the Product Line Adequacy of Existing Components", *1st Workshop on Reengineering Towards Product Lines (RIPL'05)*, 2005.
- [17] K. Schmid and M. Verlage, "The Economic Impact of Product Line Adoption and Evolution", *IEEE Software*, 19(4), July/August 2002, pp. 50-57.
- [18] K. Schmid, "A Comprehensive Product Line Scoping Approach and Its Validation", in *Proceedings of the 24th International Conference on Software Engineering (ICSE'02)*, May 2002, pp. 593-603.