Goal Model Integration for Tailoring Product Line Development Processes

Arfan Mansoor
Software Architectures and
Product Line Group
Ilmenau University of Technology
Ilmenau, 98693, Germany

Detlef Streitferdt
Software Architectures and
Product Line Group
Ilmenau University of Technology
Ilmenau, 98693, Germany

Muhammad Kashif Hanif
Department of Computer Science
Government College University
Faisalabad, 38000, Pakistan

Abstract—**Many companies rely on the promised benefits of product lines, targeting systems between fully custom made software and mass products. Such customized mass products account for a large number of applications automatically derived from a product line. This results in the special importance of product lines for companies with a large part of their product portfolio based on their product line. The success of product line development efforts is highly dependent on tailoring the development process. This paper presents an integrative model of influence factors to tailor product line development processes according to different project needs, organizational goals, individual goals of the developers or constraints of the environment. This model integrates goal models, SPEM models and requirements to tailor development processes.**

Keywords—Goal model; Product Line; Development Process; Process Line

I. Introduction

Many companies rely on the promised benefits of product lines, targeting systems between fully custom made software and mass products. Such customized mass products account for a large number of applications automatically derived from a product line. This results in the special importance of product lines for companies with a large part of their product portfolio based on their product line. The success of product line development efforts is highly dependent on tailoring the development process. This paper presents an integrative model of influence factors to tailor product line development processes according to different project needs, organizational goals, individual goals of the developers or constraints of the environment. The model integrates goal models, SPEM models and requirements to tailor development processes.

Software systems developed based on the product line approach result in systems between custom made software and systems developed for a mass market. Thus, software product lines are customized mass products. The architecture of a product line consists of a core and diverse variable components. Any members of a product line are based on its core and one or more variable components. Core and variable components are pre-developed what results in the special usage of a product line. The customer simply selects and may parametrized the desired features of the future system. Based on the product line, the system (in more detail, the software application) will be automatically generated. The effort for the development of a product line core and its variable components will reach a break even point starting from four [1] up to five [2] sold applications. This is mainly due to the large development efforts for the core of the product line, the product line training needed for the developers, the migration effort for companies to go towards the product line concept and the process maturity level needed for product line development [3]. The efforts for product line specific development processes are higher than the efforts for the development of standard systems and such development processes need to be tailored towards the project environment of the development team [4], [5]. The survey of 273 software projects in[6] revealed a potential of reducing the development effort up to 21% by raising the CMM level by one. This shows the big potential of defined and tailored development processes. For the remainder of this paper the terms method and process are used according to the Software & Systems Process Engineering Metamodel (SPEM) of the Object Management Group (OMG). A method is a reusable and goal oriented procedure made of several steps, referred to as tasks. A process is a sequence of tasks together with the timing information for the sequence. Thus, a process would contain all the timed steps needed to develop a product line. As an example, a review is taken from the method library and reused at different occasions in the process to validate the documents developed along the product line development process. Ten product line case studies have been analysed in [2] out of the domains embedded, oil and gas, finances, mobile communications, telecommunications, multimedia, and the medical domain. All the case studies use a twofold development process, with a domain engineering (development of the product line itself) and an application engineering (development of applications based on the product line) phase, as shown in figure 1. Both phases are further subdivided in a requirements, a design, a realization and a testing phase. The common assets, managed in a repository, are in between both phases. They are developed in the domain engineering phase and used in the application engineering phase.

The challenges are the development methods and processes, which have been individually and manually defined by all case studies in [2] as the project proceeded. Although guidelines for the development of product lines have been developed [2], detailed recommendations for the tailoring step of a development process are still missing. It is not yet clear whether and to what degree a given development process will

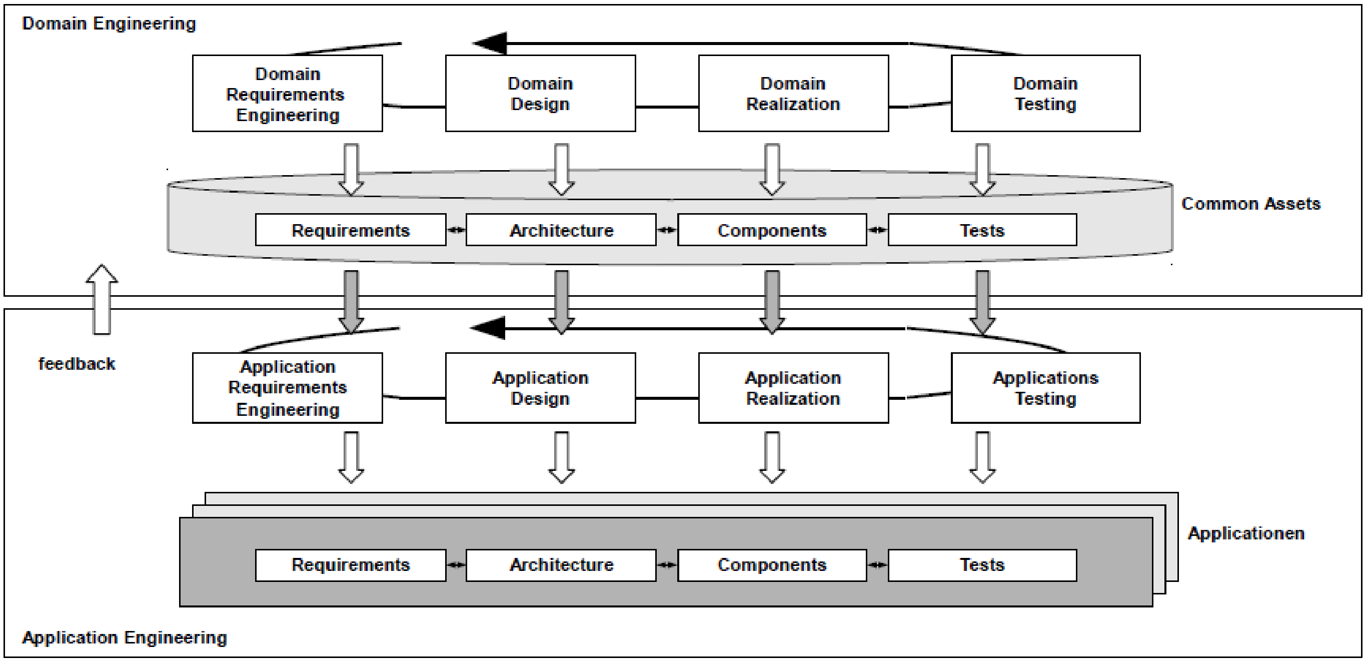

Fig. 1: Product Line Development Process

fit to its development environment. A structured approach to address this savings potential could be defined attributes together with a model to optimize the tailoring step of the development process for product lines. Therefore, a tailoring meta-model with a set of attributes to enhance the tailoring step with an optimization towards the presented attributes is presented.

The rest of the paper is organized as follow: section II discusses the need of integration model where the requirements on tailoring product line development processes are manifold. In section III requirements are divided into two main parts and the factors effecting the tailoring development processes are determined. Section IV presents a meta-model for tailoring development processes and a pseudo-code for the selection of elements. Finally, last section concludes the paper.

II. The Need of Integration Model

The product line development method PuLSE as presented in [7] is equipped with the PuLSE Baselining and Customization (PuLSE-BC) [4] procedure to tailor PuLSE towards the needs of an organization. Any tailoring decisions are bound to the variable parts of the development process. The criteria for tailoring are based on organizational and project domain issues. Such manually elicited criteria result in the variability of the development process. Pulse-BC is managing this variability in an own model. A further refinement of tailoring product line development processes is presented in [8]. Here, a product line for development processes is proposed, referred to as process line. The requirements of the development processes in this process line are based on an analysis of current and future products, projects and processes. Thus, the processes are optimized towards the products and projects, to derive a tailored development process based on the process line. Tailoring is realized with prioritized attributes, with which the resulting elements of the product, process and project analysed are ranked. An automated analysis of the underlying models is not yet realized what also hinders the efficient analysis of different scenarios in different domains. The company specific strategy and the goals of groups as well as individual developers, referred to as soft attributes are also missing. Nevertheless such attributes are important since personal factors influence the success of development process changes to a larger degree than technological challenges [9], [10],[11]. As a result, a process line model based on products, processes and project data in relation to models of the company strategy and developer goals is needed. Here, the relations of the model elements and features of the process line are highly important to be able to realize its variability [12]. In addition, there is also need of a complete model of the attributes to enable an enhanced assessment of derived development processes. Development process like the V-Model XT, SCRUM or OpenUP are targeting single system development efforts. Nonetheless parts of the methods are taken for the product line development. In [13] parts of an agile development process have been used for the product line development in a large company (SAP). Again, tailoring of development processes for product lines is an important success factor. As described in [13] but not yet accomplished, the strategic and business goals of an organization need to be part of the development process. The selection of process steps should be traceable to the business and strategic goals. Without such traces development processes cannot be fully analysed and tailored. Thus, the business goals need to be part of the above described process line. In [14] the tailorability of the V-Model XT towards product line development is analysed. Based on this work a process line was developed and a V-Model XT development processes could be derived based on the process line. Unfortunately, the selection of supporting tools for the development process is still left to the project manager and/or developer and the selection of tools is bound to the knowledge about their advantages and drawbacks, what is currently not part of the model of process lines. The analysis of product line approaches emphasizes the relevance of tools for the success of a product line development project. All the presented approaches in this paper are based on the product line development concept shown in figure 1 and offer ideas to relate the development process to the development environment. Although, none of the approaches is able to offer a complete model of a tailorable development process together with the elements/components of the development environment. Here, the analysis and assessment of development processes need to include tools, since they strongly influence the expected effort of a product line development project. The relation of decisions to the original goals of the decisions can be realized with goal models [15]. Goal oriented business processes with variabilities are presented in [16]. Such models could be used as in [17] to analyse and assess the chances of success with the Goal-Question-Metric (GQM) method for product line development projects. For the tailoring step of a developers environment the in influential factors and attributes are still missing for process lines, but could be realized using a goal model. Thus, a comprehensive view onto product line development domain would be possible. Finally an integrative model for the description of stakeholder needs and goals in relation to the development process artifacts and the development environment specifics is needed, to be able to analyse potential influences of changing goals early in the project development.

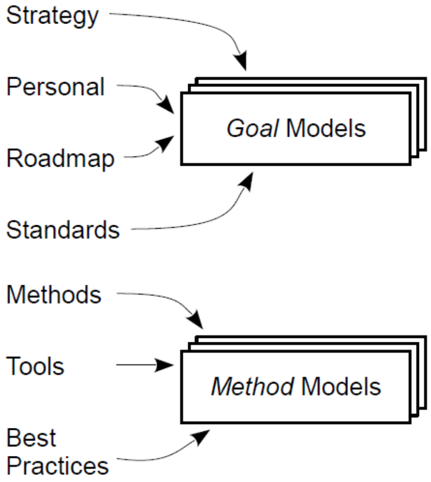

Fig. 2: Goal and Method Models

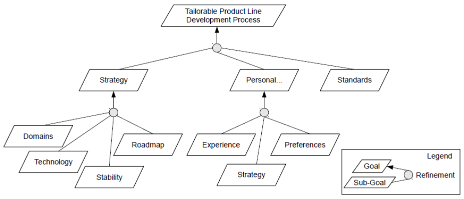

Fig. 3: Integrated Goal Model

III. Tailoring Development Processes

As stated in the previous section the requirements on tailoring product line development processes are manifold. Here, these requirements are divided in two main parts

1) The goal model based requirements
2) The method model based requirements

The following categories and parts of the two models are based on own experiences in industrial projects and lessons learned within student software development projects. First, the identification of influence factors that can be described by goal models contains soft factors, as shown in figure 2.

Based on experience, it is estimated that about 70% of the challenges throughout the software development project can be traced back to such soft factors. Thus, addressing such factors can influence the success of a project by a large degree. As shown in figure, 3, two top level factors are refined with a goal model.

The **strategy**of a company is very important when comes to the initial decision for or against a product line. Thus, the following sub-goals as refinement of the strategy are tightly connected to the product line development.

- The target **domain** or domains of the products that will be developed rule about the product line approach. New domains or domains that will be abandoned in the future need to be known and elicited in the requirements engineering phase. Of course, these requirements might have a large impact on the architecture of the product line, specifically to the core and the variabilities of the product line.
- Any strategic choice of the **technology** influences the future constraints (performance, memory, available development environment, available compilers) of the system and thus, constraints for the product line. For example, the realization of variabilities with the C language has reduced capabilities compared to C++.
- **Stability** of the strategy. For new companies this is highly relevant. The strategy is subject of a high risk for changes. Thus, this goal influences the overall feasibility of the product line development.
- The **roadmap** includes the timing for the release of product features. For each release a set of features is identified. The length (way into the future) of the roadmap influences the technological choices and the re-development of the product line. Due to technological changes, fluctuation of employees (and with them the knowledge) and unforeseen requirements the implementation of the architecture of a product line needs to be adapted to this new environment. The roadmap needs to address these large and periodic updates.

The personal factors also have a large impact onto the other elements in the goal model. The personal goals are coupled with a stakeholder model of the involved persons in a software project. Each stakeholders should have an own personal goal model reflecting his/her position towards the product line

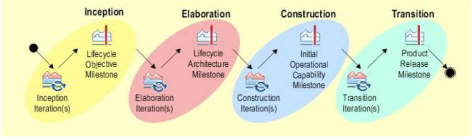
Inception
Elaboration
Construction
Transition
Lifecycle Objective Milestone
Lifecycle Architecture Milestone
Initial Operational Capability Milestone
Product Release Milestone
Inception Iteration(s)
Elaboration Iteration(s)
Construction Iteration(s)
Transition Iteration(s)

Fig. 4: OpenUP Overview

development process. Since this is a very personal information it is recommended to keep this model private but use the information in correlation with the other models (strategy and standards) as well as use the results of a model analysis as input for the periodic discussions with the management within the company and/or of the respective project.

- Each stakeholders own experience should be related to the role descriptions of the basic development processes (e.g., OpenUp, SCRUM). Besides the potentials for further personal development, such an experience level should be related to the project roles (and their skills) which are attached to each development step. For exchangeable development steps, experiences set the rules on which step to take.
- Each stakeholder has preferences for application domains or technological choices. There are also preferences for methods used along the development process or for specific templates to be used for the deliverables of the development process. These preferences will influence the choices of the method and development process parts of the product line.
- Each stakeholder might (or should) have an own strategy in contrast to the company strategy. The alignment of the strategy of all different stakeholders is impossible, due to the private nature of this information. As with the experience, the awareness of the other goals and their correlation to the own strategy is an important step towards the integration into a developer group and a good starting point to develop an own roadmap. The individual analysis of the own strategy is a good point to think about the own position in the company and/or to better understand the own position.

Standards will influence the technology goals for the strategic planning and they recommend or require technologies and/or tools. For example, the safety standard IEC61508 recommends test case generation tools. Standards could also require a specific development process structure and give recommendations or require development methods. The lower part of figure 2 shows the method models. Here, we use SPEM to describe all the needed parts of the methods, processes and best practices. As a SPEM implementation, OpenUP is shown in figure 4.

OpenUP is an open source development process for standard applications, the complete extension of OpenUP towards a product line is a future work package. Nevertheless this process is taken as tailoring example to address the above mentioned goals. The development process is split into four iterative phases. Compared to figure 1, the requirements is equivalent to the inception phase, the design is equivalent to the elaboration phase and the realization is equivalent to the construction phase. The testing steps are present in each iteration of the OpenUP process and at the first sight the testing phase in figure 1 does not match the OpenUP transition phase, but this testing phase is meant to be the final system test with an iterative testing approach as well and thus, the two models are comparable. For each of the development steps in figure 4 parts of the method steps of the OpenUP method library are taken and put together. Each task has its responsible roles attached and each role has its tasks attached. As shown in figure 5 the developer role is required to perform the five given tasks and is also responsible for the four deliverables. The last of the SPEM elements relevant for the process tailoring step are the guidances. As shown in figure 6 there are 14 guidance types which can be used to support any SPEM element, e. g., a task.

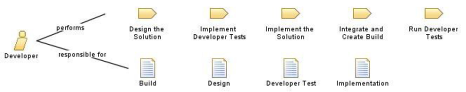

Fig. 5: Developer Role in OpenUp

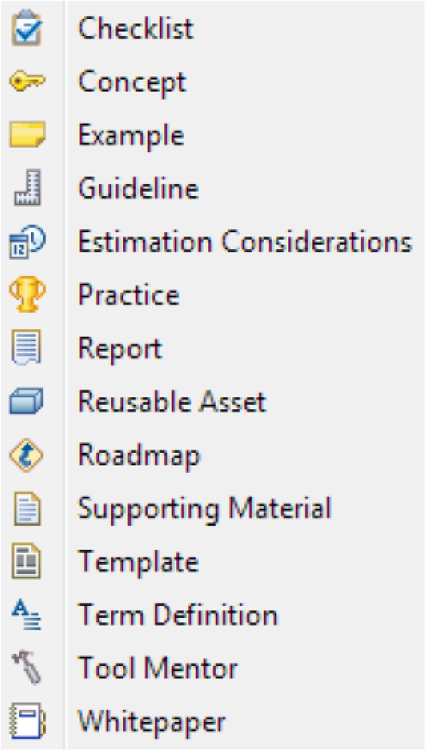
Checklist
Concept
Example
Guideline
Estimation Considerations
Practice
Report
Reusable Asset
Roadmap
Supporting Material
Template
Term Definition
Tool Mentor
Whitepaper

Fig. 6: OpenUP Guidance for SPEM elements

IV. Tailoring Meta-model

Based on the above mentioned relations between goal models, method/process models and requirements, the proposed meta-model as shown in figure 7. The **Element** abstracts the **Goal** model elements, the **MethodElements** of SPEM, and the **Requirement** elements found in most of the meta-models of requirements management tools like Polarion. The meta-model now allows to connect any element using links of the abstract

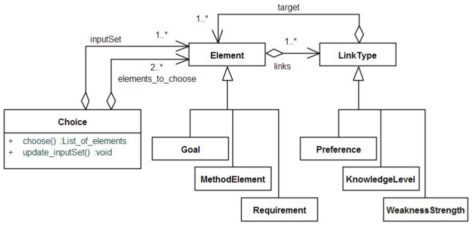

Fig. 7: Meta-model for Development Process Tailoring

```
ERMAP of elements_to_choose,rank
for all A in inputSet do
    $linkSubSet =$
    $getLinkTypesFromTo(A, elements\_to\_choose)$
    $adjustRank(linkSubSet)$
end for
$select\_SPEM\_based(ermap)$
```

Fig. 8: choose Pseudo-code

LinkType. Currently the following link types are defined:

- Preferences - Are used to indicate a stakeholders preference for a given element (e. g., a developer might have a preference for a text editor which is part of the guidances of the process model). The preference link can have values between -100% (aversion against an element) up to +100% (this element is vitally important for a stakeholder)
- KnowledgeLevel - This link indicates the level of confidence a stakeholder might have with an element in our model. The knowledge level link is divided in two categories. The knowledge as user of an element between 0% (the stakeholder knows nothing about an element) and 50% (the stakeholder knows everything to use and work with an element). The knowledge as teacher for an element my have values between 51% (the stakeholder has taught the use of an element at least once) and 100% (the stakeholder is an experienced teacher with more than 5 years of teaching experience).
- WeaknessStrength - Any element might weaken or strengthen another element. For example, the presence of a requirement for safety in the medical domain will result in high documentation demands what in consequence will strengthen the quality of the final product and at the same time weaken a fast delivery of the product. The weakness/strength link can have values between -100% (the source element will disable/weakens the target element) up to +100% (the source element requires/strengthens the target element. Thus, the target element be comes mandatory)

To work with the product line approach, variabilities are needed, as discussed in the first sections. The variability of the process is modelled with the SPEM content variability types (contributes, extends, replaces, extends and replaces) for the elements of a SPEM model. To trigger this variability of the process model, the **Choice** is introduced in the tailoring meta-model in figure 7. This has an input set of elements influencing the choice. This input set will be updated by the ***update_inputSet()*** method whenever the choices are going to be evaluated. This method will search for elements with target links present in the ***elements_to_choose*** list and will update the ***inputSet*** list accordingly. Once the ***update_inputSet()*** method has been executed the ***choose()*** method can follow with its execution to calculate the variant based on the given input elements.

The pseudo-code in figure 8 shows how to calculate the choice of elements. First a map of elements and its ranking is created.

For all the elements in the list of input elements, the elements which have links to elements in the ***elements_to_choose*** list are filtered out . This is accomplished by the ***getLinkTypesFromTo*** method which stores its results in a list of links as subset of the original ***links*** list of the ***Element*** type. This list is then taken as input for the ***adjustRank*** method which in the current version simply adds the values for the preferences, knowledge level and weakness/strength values, to the ***ermap*** rankings discussed in the last section. Finally, a selection of choices based on the rankings and the SPEM models constraints is made. This meta-model can be extended in two ways:

1) First, any additional elements can be added to this meta-model to address future models which need to be integrated in the tailoring process.
2) Second, the link types can be extended by new links needed in the future.

V. Conclusion

In this paper, the current state of the product line development domain and the challenges are discussed when it comes to the development processes which need to be adapted to the specific needs of the development teams. Tailoring product line development processes has been identified to enable large savings for the domain engineering as well as application engineering phase of product line development projects. For an integrative approach to process line tailoring, a tailoring meta-model is proposed which includes goal models, SPEM process models as wells as requirements. With this model stakeholder specific goals can be used to support binding a variable part of the development process. This support addresses soft factors as well as concrete requirements. Future research work will be spent to further elicit attributes of different domains influencing the development process. In addition the enhancement of the few variable process steps in OpenUP towards a complete process line will also be subject of future research efforts.

VI. Acknowledgment

We acknowledge support for the Article Processing Charge by the German Research Foundation and the Open Access Publication Fund of the Technische Universit Ilmenau.

References

[1] D. M. Weiss and C. T. R. Lai, *Software Product-line Engineering: A Family-based Software Development Process*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

[2] F. J. v. d. Linden, K. Schmid, and E. Rommes, *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[3] L. Brownsword and P. Clements, “A case study in successful product line development,” Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU/SEI-96-TR-016, 1996. [Online]. Available: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=12587

[4] K. Schmid and T. Widen, “Customizing the pulse$^{\text{tm}}$ product line approach to the demands of an organization,” in *Software Process Technology, 7th European Workshop, EWSPT 2000, Kaprun, Austria, February 21-25, 2000, Proceedings*, 2000, pp. 221–238. [Online]. Available: http://dx.doi.org/10.1007/BFb0095031

[5] V. R. Basili and H. D. Rombach, “Tailoring the software process to project goals and environments,” in *Proceedings of the 9th International Conference on Software Engineering*, ser. ICSE ’87. Los Alamitos, CA, USA: IEEE Computer Society Press, 1987, pp. 345–357. [Online]. Available: http://dl.acm.org/citation.cfm?id=41765.41804

[6] B. K. Clark, “The effects of software process maturity on software development effort,” 1997.

[7] J. Bayer, O. Flege, P. Knauber, R. Laqua, D. Muthig, K. Schmid, T. Widen, and J.-M. DeBaud, “Pulse: A methodology to develop software product lines,” in *Proceedings of the 1999 Symposium on Software Reusability*, ser. SSR ’99. New York, NY, USA: ACM, 1999, pp. 122–131. [Online]. Available: http://doi.acm.org/10.1145/303008.303063

[8] O. Armbrust, M. Katahira, Y. Miyamoto, J. Münch, H. Nakao, and A. Ocampo, “Scoping software process lines,” *Softw. Process*, vol. 14, no. 3, pp. 181–197, May 2009. [Online]. Available: http://dx.doi.org/10.1002/spip.v14:3

[9] M. Inoki and Y. Fukazawa, “Software product line evolution method based on kaizen approach,” in *Proceedings of the 2007 ACM Symposium on Applied Computing*, ser. SAC ’07. New York, NY, USA: ACM, 2007, pp. 1207–1214. [Online]. Available: http://doi.acm.org/10.1145/1244002.1244266

[10] D. Stelzer and W. Mellis, “Success factors of organizational change in software process improvement,” *Software Process: Improvement and Practice*, vol. 4, no. 4, pp. 227–250, 1998. [Online]. Available: http://dx.doi.org/10.1002/(SICI)1099-1670(199812)4:4¡227::AID-SPIP106¿3.0.CO;2-1

[11] M. Niazi, D. Wilson, and D. Zowghi, “Critical success factors for software process improvement implementation: an empirical study,” *Software Process: Improvement and Practice*, vol. 11, no. 2, pp. 193–211, 2006. [Online]. Available: http://dx.doi.org/10.1002/spip.261

[12] D. Benavides, S. Segura, and A. Ruiz-Cortés, “Automated analysis of feature models 20 years later: A literature review,” *Inf. Syst.*, vol. 35, no. 6, pp. 615–636, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.is.2010.01.001

[13] B. Blau and T. Hildenbrand, “Product line engineering in large-scale lean and agile software product development environments - towards a hybrid approach to decentral control and managed reuse,” in *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*, Aug 2011, pp. 404–408.

[14] T. Ternite, “Process lines: A product line approach designed for process model development,” in *Software Engineering and Advanced Applications, 2009. SEAA ’09. 35th Euromicro Conference on*, Aug 2009, pp. 173–180.

[15] A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley, 2009. [Online]. Available: http://books.google.de/books?id=AYk_AQAAIAAJ

[16] E. Santos, J. Castro, J. Sánchez, and O. Pastor, “A goal-oriented approach for variability in BPMN,” in *Anais do WER10 - Workshop em Engenharia de Requisitos, Cuenca, Ecuador, April 12-13, 2010*, 2010.

[17] B. Geppert and D. M. Weiss, “Goal-oriented assessment of product-line domains,” in *Software Metrics Symposium, 2003. Proceedings. Ninth International*, Sept 2003, pp. 180–188.