Improving Product Derivation in Software Product Line Engineering

Pádraig O’Leary, Ita Richardson
Lero – the Irish Software Engineering Research Centre
University of Limerick, Ireland
{ padraig.oleary | ita.richardson}@lero.ie

Steffen Thiel
Department of Computer Science
Furtwangen University of Applied Sciences
Germany
steffen.thiel@hs-furtwangen.de

Abstract—The derivation of individual products from a software product line is still seen as a time-consuming and expensive activity in many organisations. Despite recognition that an effective derivation process could alleviate many of the difficulties associated with product development, little work has been dedicated to this area. Existing approaches have very different scope and emphasize various aspects of the derivation process. Furthermore, they are frequently too specialised to a particular development technique to serve as a general solution. This leaves organisations with no centralized starting point for defining an approach to product derivation.

Accordingly there is a strong need for a structured approach to product derivation which defines activities, tasks, roles, inputs and outputs of each step in a systematic way. Through a series of research phases using sources in industry and academia, this research has developed a process reference model for product derivation (Pro-PD) which is briefly sketched in this paper. Pro-PD focuses on the essential tasks, roles and work artefacts used to derive products from a software product line. The model is also adaptable and can be tailored to suit different requirements.

Keywords—product derivation; process model; software product line engineering; activities; artefacts; stakeholders

I. INTRODUCTION

A Software Product Line (SPL) is a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [1]. The SPL approach makes a distinction between domain engineering, where a common platform for a number of products is designed and implemented, and application engineering, where a product is derived based on the platform components [2]. The separation into domain engineering and application engineering allows the development of software artefacts which are shared among the products within that domain. These shared artefacts become separate entities in their own right, subscribing to providing shared functionality across multiple products.

It is during application engineering that the individual products within a product line are constructed. The products are built using a number of shared software artefacts created during domain engineering. The process of creating these individual products using the platform artefacts is known as product derivation.

Product Derivation is the process of constructing a product from a Software Product Line’s (SPL) core assets [3]. An effective product derivation process can help to ensure that the benefits delivered through using these shared artefacts across the products within a product line is greater than the effort required to develop the shared assets. In fact, the underlying assumption in SPL that “the investments required for building the reusable assets during domain engineering are outweighed by the benefits of rapid derivation of individual products” [4] might not hold if inefficient derivation practices diminishes the expected gains.

II. PROBLEM STATEMENT

In the context of inefficient product derivation, a number of publications speak of the difficulties associated with the process. Hotz et al. [2] describe it as “slow and error prone, even if no new development is involved”. Griss [5] identifies the inherent complexity and the coordination required in the derivation process by stating that “…as a product is defined by selecting a group of features, a carefully coordinated and complicated mixture of parts of different components are involved”. Therefore, the derivation of individual products from shared software assets is still a time-consuming and expensive activity in many organisations [3].

Despite this, there has been little work dedicated to the overall product derivation process. Rabiser et al. [6], for example, claim that “guidance and support are needed to increase efficiency and to deal with complexity of product derivation”. Deelstra et al. [3] state that there “is a lack of methodological support for application engineering and, consequently, organizations fail to exploit the full benefits of software product families.” As a result current approaches fail to provide a holistic view of product derivation leaving organizations with no centralized starting point for defining an approach to product derivation. This results in ad-hoc solutions.

Due to this need for a structured approach to product derivation, the authors identified the following research objective:

To define a systematic process, which will provide a structured approach to the derivation of products from a software product line, based on a set of tasks, roles and work products, and to demonstrate its adaptability to different process models.
To meet this objective, we developed Pro-PD: Process Model for Product Derivation. Pro-PD was iteratively developed and evaluated through four research phases involving academic and industrial sources.

III. APPROACH

In order to achieve the objective defined above, this research has developed the Pro-PD process for product derivation. Pro-PD is a process reference model for product derivation that is minimal, complete, and adaptable:

- **Minimal** – only content that is seen as essential for product derivation is included
- **Complete** – it can be manifested as an entire process to build a system
- **Adaptable** – it can be adapted to different process types

Pro-PD is a minimally complete process reference model for product derivation. This means that only fundamental product derivation process content is included. Domain and discipline specific content is not included in Pro-PD and Pro-PD is independent of the methods and techniques used to derive a product. Pro-PD focuses instead on the essential tasks, roles and artefacts used to derive products from a software product line.

Pro-PD is adaptable; it can be used as a foundation from which company specific product derivation process content can be developed. The process structure is based on the waterfall model [7]; however, to demonstrate its flexibility, it is adapted to fit the characteristics of an iterative process model.

IV. RESEARCH DESIGN

The objective of our research is to provide an evidence based process approach for product derivation. With this in mind, our research design was influenced by Ahlemann et al. [8] which focused on empirically grounded and valid process model construction. Phase 1 and 2 are the primary construction steps. Phase 3 is both a development and an evaluation step. Phase 4 is purely an evaluation step. An overview of the research design is presented in Figure 1.

Pro-PD was developed using development – evaluation phases (see Figure 1). Version one of Pro-PD was developed through sources in the literature and captured expert opinion, and evaluated during an industrial case study with Robert Bosch GmbH. Version two was developed based on a study of derivation practices at Robert Bosch GmbH and evaluated during a research collaboration with the DOPLER laboratory. Version three was developed based on the findings of the research collaboration with DOPLER. This version was evaluated through an inter-model evaluation and an adapted SPL evaluation framework. Finally, with the results of this closing evaluation integrated, version four of Pro-PD was developed.

![Figure 1. Research Design](image-url)
V. PRO-PD OVERVIEW

Pro-PD focuses on the roles, artifacts, tasks and activities used to derive products from a software product line. These elements represent the process building blocks of Pro-PD. Roles represent a set of related skills and responsibilities. Artifacts are produced, modified or used by tasks. Tasks are assignable units of work that usually consume or produce one or more products. Activities are grouping of related tasks that share a specific development goal.

A. Units of Work: Tasks and Activities

Tasks are assignable units of work that consume or produce one or more products. To make process building and understanding easier, Pro-PD groups related tasks into Activities. Activities have specific development goals. Pro-PD uses activities as the building blocks from which different development phases (i.e. Preparing for Derivation, Product Configuration, Product Development and Testing) can be constructed during process instantiation. Figure 2 gives an overview of these Pro-PD activities and the iterative nature of the Pro-PD process.

These activities can be described as:

- **Initiate Project** - the preparatory tasks required to establish a product derivation project.
- **Identify and Refine Requirements** – the preparatory tasks required to commence a new iteration of the product derivation project.
- **Derive the Product** - creates an integrated product configuration that makes maximum use of the platform and minimises the amount of product specific development required.
- **Develop the Product** - facilitates requirements that could not be satisfied by a configuration of the existing assets through component development or adaptation.
- **Test the Product** - validates the current product build.
- **Management and Assessment** - provides feedback to the platform team and monitor progress of derivation project.

Table 1 lists the tasks performed for each of these activities.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Tasks performed in this activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiate Project</td>
<td>Translate Customer Requirements; Coverage Analysis; Customer Negotiation; Create the Product Requirements</td>
</tr>
<tr>
<td>Identify and Refine</td>
<td>Find and Outline Requirements; Create the Product Test Cases; Allocate Requirements; Create Guidance for Decision Makers</td>
</tr>
<tr>
<td>Derive the Product</td>
<td>Select Closest Matching Configuration; Derive New Configuration; Evaluate Product Architecture; Select Platform Components; Product Integration; Integration Testing; Identify Required Product Development</td>
</tr>
<tr>
<td>Develop the Product</td>
<td>Component Development; Component Testing; Product Integration; Integration Testing</td>
</tr>
<tr>
<td>Test the Product</td>
<td>Run Acceptance Tests</td>
</tr>
<tr>
<td>Management and Assessment</td>
<td>Provide Feedback to Platform Team, Monitor Project</td>
</tr>
</tbody>
</table>

Figure 2. Overview of Prod-PD Activities
B. Roles

Despite attempts to automate product derivation, it remains a human activity in which tasks are performed through collaboration and the exchange of work. In Pro-PD there are several roles that represent the different responsibilities, which occur during product derivation. These roles are: Customer, Platform Manager, Product Architect, Product Developer, Product Manager and Product Tester. These roles are assigned to specific tasks, which create and modify the different work products.

Table 2 lists specific roles and responsibilities of Pro-PD.

VI. CONCLUSION

This research has resulted in a number of contributions. An extensive literature review is presented that highlights issues within current approaches to product derivation. Observations on product derivation practice from both academia and industry are described. An adaptable process model for product derivation is developed and evaluated. The process can be seen as a systematic way to perform product derivation through a well-defined sequence of activities, tasks, roles and artifacts. The research also contributes to the research area through the publication of the knowledge gathered during the research and the presentation of the research to SPL groups, both academic and industrial, in Luxembourg, Germany, Ireland, and Brazil.

Even it being an important contribution to the field, this work into a comprehensive process for product derivation is not complete. Further work is required for the establishment of process support for product derivation. The researchers believe this work is a first step in this long and complex road.

ACKNOWLEDGMENTS

This research has been funded by Science Foundation Ireland through IRCSET under grant no. RS/06/167.

### Table II. Pro-PD Roles and Responsibilities

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform Manager</td>
<td>Represents the interests of the platform during the derivation project. The role should have a degree of understanding on the demands of the product team.</td>
</tr>
<tr>
<td>Product Architect</td>
<td>Responsible for the major technical decision making within the derivation project. The role requires a good knowledge of the platform and an understanding of the demands on the platform team.</td>
</tr>
<tr>
<td>Product Developer</td>
<td>Responsible for Component Development and Component Testing. The Product Developer needs to be able to understand and conform to the product architecture.</td>
</tr>
<tr>
<td>Product Manager</td>
<td>Responsible for customer relationship management, negotiation of product features with the customer and project planning.</td>
</tr>
<tr>
<td>Product Tester</td>
<td>Responsible for the main testing effort within the project. The Product Tester should co-ordinate with the platform testing team to reuse Platform Test Artefacts.</td>
</tr>
</tbody>
</table>

### Table III. Pro-PD Artefacts

<table>
<thead>
<tr>
<th>Artefact Type</th>
<th>Artefacts</th>
</tr>
</thead>
</table>

REFERENCES


