Information and Software Technology 58 (2015) 319–337

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

# Product derivation in practice

Leandro Oliveira de Souza [a,b,d,*], Pádraig O'Leary [e,f], Eduardo Santana de Almeida [c,d], Sílvio Romero de Lemos Meira [b,d]

[a] Federal Institute of Bahia (IFBA), Rodovia BA 148, Km 04, 1800, Vila Esperança, Irecê, BA 44900-000, Brazil
[b] Federal University of Pernambuco (UFPE), Recife, PE, Brazil
[c] Federal University of Bahia (UFBA), Salvador, BA, Brazil
[d] Reuse in Software Engineering (RiSE), Recife, PE, Brazil
[e] Lero – The Irish Software Engineering Research Centre, University of Limerick, Ireland
[f] ARCH – Applied Research in Connected Health, University of Limerick, Ireland

ARTICLE INFO

*Article history:*
Received 12 March 2013
Received in revised form 3 July 2014
Accepted 5 July 2014
Available online 19 July 2014

*Keywords:*
Case study
Multiple case study
Constant comparison analysis
Cross-case analysis
Product derivation
Software product lines

ABSTRACT

*Context:* The process of constructing a product from a product line of software assets is known product derivation. An effective product derivation process is important in order to ensure that the efforts required to develop these shared assets is lower than the benefits achieved through their use. Despite its importance, relatively little work has been dedicated to the product derivation process and the strategies applied in practice. Additionally, there is a lack of empirical reports describing product derivation in industrial settings, and, in general, where these reports are available, they have been conducted as informal studies.
*Objective:* Our aim is to investigate how product derivation is performed in practice.
*Method:* We apply a multi-case study design to two different industrial software product line projects with the goal of investigating how they derive their products in practice. The findings from our studies were individually analyzed using the *Constant Comparison* technique. In order to identify patterns across these studies, the findings were compared using a *Cross-case* analysis approach.
*Results:* The research approach allowed us to examine the case study outcomes from different perspectives, capturing similarities and differences. From the cases, we identified context specific strategies for product derivation which are easier for practitioners to contextualise and implement.
*Conclusions:* The case studies provide method-in-action insights into concepts explored in the literature, such as: iterative and incremental product derivation, instantiation and integration of platform components and derivation of product databases. Practitioners can use this work as a basis for defining, adapting or evaluating their own product derivation approaches. While researchers can use this work as a starting point for new industrial reports, presenting their experiences with product derivation.


## 1. Introduction

Software reuse is an important approach for companies interested in productivity gains, minimizing development costs and time-to-market improvements [1]. However, these benefits are not assured by the application of ad hoc reuse approaches. These are generally considered opportunistic reuse, not systematic, and typically restricted to the code level.

In this context, *Software Product Lines (SPL)* emerges as a systematic way to achieve reuse [2]. It applies a strategy that plans the use of assets in multiple products rather than ad hoc approaches that reuse assets only if they happen to be suitable [3].

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 [4]. The separation into domain engineering and application engineering enables the distinction between the platform building process and specific-product building process. However, these processes must interact in a way that is beneficial to both.

The process of creating individual products during application engineering is known as *Product Derivation (PD)*. It is the practice of constructing a product from a product line of software assets that is developed using shared product family artifacts [5]. In a product line organization, the use of an effective product

* Corresponding author at: Federal Institute of Bahia (IFBA), Rodovia BA 148, Km 04, 1800, Vila Esperança, Irecê, BA 44900-000, Brazil. Tel.: +55 74 8841 3910.
*E-mail addresses:* leandro.souza@ifba.edu.br (L.O. de Souza), padraig.oleary@ul.ie (P. O'Leary), esa@rise.com.br (E.S. de Almeida), srlm@cin.ufpe.br (S.R. de Lemos Meira).

http://dx.doi.org/10.1016/j.infsof.2014.07.004
0950-5849/

derivation process can help to ensure the return on investment required to develop the platform assets [6]. *"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"* [5] might not hold if inefficient derivation practices diminishes the expected gains.

In the SPL context, an effective product derivation process is important in order to ensure that the effort required to develop the platform assets will be lower than the benefits achieved through using these shared assets by products belonging to products line. However, the area of product derivation is still rather immature [7]. Other than a high level description of the activities required there is a lack of support for the derivation process [6]. In comparison with the research in developing and modeling product lines, few approaches and tools are available for product derivation [7]. Additionally, existing approaches do not present detailed information on the strategies for product customization, resolving variability, and the derivation of database models.

In addition, there are few case studies [8,5] available describing how SPL organizations derive products from a product line. The documented studies have been conducted as informal case studies without sufficient attempt to provide scientific rigor with empirical research methods, making their replication hard.

Thus, the issues associated with product derivation demonstrate a strong need for more research to improve understanding of the process and identify efficiencies. We believe that this can be achieved through empirical studies, which provide reports on industrial experience describing details about how software organizations derive their products from shared product family artifacts.

In this context, we used a multiple-case study design [9] in two different industrial SPL projects to provide evidence on how product derivation is performed in SPL companies. The first study was conducted at Company A which develops integrated management systems for the medical domain since 1994. Then, the case study was replicated at Company B which develops integrated systems for the management and operational control of complementary social security entities since 1996. Their findings were individually analyzed based on the *Constant Comparison* analysis technique [10,11]. From these findings, we performed a "*Cross-case*" [12] analysis between both cases in order to identify similarities and differences across their findings. These multiple-case studies were influenced by [9] and based on the guidelines defined in [13,14].

The remainder of this paper is organized as follows: Section 2 describes the contexts in which the case studies were performed. Section 3 presents the research methodology, including the research design, research questions, selection of subjects, data collection procedures and analysis. In Sections 4 and 5, the results of the multiple-case studies are presented. In Section 6, we compare the identified activities with *Key Activities for Product Derivation in SPL* [15] and discuss the results. Section 7 discusses the findings achieved from the *cross-case* analysis. Section 8 presents the validity procedures and threats to the study. Section 9 discusses related work on product derivation. Finally, conclusions and future directions are presented in Section 10.

## 2. Case study context

This section describes the two case study contexts.

### 2.1. Company A

Company A is located in Salvador, Brazil and develops integrated management systems for the medical domain since 1994.

Currently, the company portfolio has four major products. *SmartHealth* is composed of 35 modules (or sub-domains), and manages the whole hospital area, from financial to patient aspects. *SmartClin* contains 28 modules, performing clinical management and supporting activities related to medical exams and diagnostics. *SmartLab* has 28 modules and integrates a set of features to manage labs of clinical pathology. Finally, *SmartDoctor* is a web product composed of 11 modules responsible for managing a doctor's office tasks and routines.

The Company A product line (SMART) is considered stable and is composed of 52 modules (sub-domains), including more than 918 features. It provides several possible variations among its different features and enables the instantiation of customized products within the medical domain. Thus, a company costumer can choose within the *SMART portfolio* the set of modules and features that satisfy their needs.

The products built on top of the *SMART Platform of Core Assets* are large and complex technical software systems with hundreds of features. Their infrastructure is composed of several parameter calls, which enables the selection of components and features during product derivation.

### 2.2. Company B

The case study was replicated at Company B. This company develops integrated systems for the management and operational control of complementary social security entities since 1996 in Salvador, Bahia, Brazil. Their customers are *Entities* (Foundations) responsible for managing pension funds of public and private companies employees.

Currently, the company portfolio includes nine sub-systems that incorporate features for different sub-domains within the complementary social security domain.

Company B has developed its own iterative and incremental product derivation methodology that aims to optimize the product customization and deployment process. From the platform, it is possible to instantiate different products that can be customized to accommodate different customers needs.

## 3. Case study design

Initially, a literature review was performed. The purpose of the literature review was to identify relevant concepts in the product derivation area which could guide the investigation.

The research method applied was an *embedded*, *exploratory* and *multiple-case study* [16,9]. The multiple-case study approach increases the external validity of the research through the implied "*replication*" inherent in its design [9]. According to [17] stated in [9], "*the evidence from multiple cases is often considered more compelling, and the overall study is therefore regarded as being more robust*".

The study had a flexible design [18], enabling possible real time modifications while always keeping its objectives. The research procedures were described in a case study protocol which was influenced by [14,9,13].

The protocol described the execution of the case study, which methods would be used for data collection, the techniques that would be applied during the case study, and which analysis method would be used. It was continuously updated as the plans for the case study changed.

The first version of the protocol was developed and revised with input from members of RiSE Labs.[1]

### 3.1. Research questions

This study investigated how product derivation is performed in industrial settings. The goals of the study motivated the *descriptive*

[1] http://labs.rise.com.br/.