Software Product-Line Evaluation in the Large

Robert Lindohf · **Jacob Krüger** · **Erik Herzog** ·
Thorsten Berger

Abstract Software product-line engineering is arguably one of the most successful methods for establishing large portfolios of software variants in an application domain. However, despite the benefits, establishing a product line requires substantial upfront investments into a software platform with a proper product-line architecture, into new software-engineering processes (domain engineering and application engineering), into business strategies with commercially successful product-line visions and financial planning, as well as into re-organization of development teams. Moreover, establishing a full-fledged product line is not always possible or desired, and thus organizations often adopt product-line engineering only to an extent that deemed necessary or was possible. However, understanding the current state of adoption, namely, the maturity or performance of product-line engineering in an organization, is challenging, while being crucial to steer investments. To this end, several measurement methods have been proposed in the literature, with the most prominent one being the Family Evaluation Framework (FEF), introduced almost two decades ago. Unfortunately, applying it is not straightforward, and the benefits

Robert Lindohf
Saab AB, Sweden
E-mail: robert.lindohf@saabgroup.com

Jacob Krüger
Otto-von-Guericke University Magdeburg, Germany & University of Toronto, Canada
E-mail: jkrueger@ovgu.de

Erik Herzog
Saab AB, Sweden
E-mail: erik.herzog@saabgroup.com

Thorsten Berger
Chalmers | University of Gothenburg, Sweden
E-mail: thorsten.berger@chalmers.se

of using it have not been assessed so far. We present an experience report of applying the FEF to nine medium- to large-scale product lines in the avionics domain. We discuss how we tailored and executed the FEF, together with the relevant adaptations and extensions we needed to perform. Specifically, we elicited the data for the FEF assessment with 27 interviews over a period of 11 months. We discuss experiences and assess the benefits of using the FEF, aiming at helping other organizations assessing their practices for engineering their portfolios of software variants.

Keywords software product lines · process maturity · experience report · family evaluation framework

1 Introduction

Software product-line engineering (SPLE) provides methods and tools to systematically reuse software assets and establish software product lines—portfolios of software products (a.k.a., variants) in an application domain. SPLE allows to mass-customize software to particular customer needs (Apel et al. 2013; Clements and Northrop 2002; Pohl et al. 2005), substantially reducing the time-to-market of new products, lowering maintenance costs, and increasing software quality (Berger et al. 2014a; Krüger and Berger 2020b; van der Linden et al. 2007). To this end, SPLE exploits the similarities and manages the variabilities of these products based on an integrated software platform. However, adopting SPLE requires substantial investments by organizations in four dimensions—*business, architecture, process, and organization*—commonly known as the BAPO concerns (America et al. 2000; van der Linden 2002; van der Linden et al. 2007).

From a *business perspective*, SPLE requires developing a strategy for the product line, developing a vision of its future use and commercial success, as well as re-organizing the financial planning. For instance, a product-oriented organization obtains revenue for concrete products, internally mapped to individual projects. However, since customers do not fund platform development, which provides value for other customers and the organization itself in the long-run, the organization needs to carefully plan the platform funding.

From an *architecture perspective*, SPLE advocates the adoption of an integrated software platform, abstractly representing and managing its assets (e.g., source code, documentation, models) using *features* (Berger et al. 2015). The platform allows to derive individual products by selecting the desired features each product should have, typically in an automated, tool-supported process. To this end, the platform incorporates so-called variability mechanisms—implementation techniques for integrating variation points in the platform, for instance, using feature toggles (a.k.a., feature flags), preprocessors, program generators or component frameworks. Since large product lines typically have thousands of variation points, controlled by features with often intricate dependencies, SPLE advocates the use of model-based representations to manage this complexity. Variability models, such as feature models (Berger et al. 2013; Czarnecki et al. 2012; Kang et al. 1990; Nešić et al. 2019), abstractly represent features and their dependencies in an intuitive, tree-like structure. As input to interactive configurator tools, feature models precisely define the valid configurations of the platform and help organizations obtain respective products.

From a *process perspective*, SPLE commonly separates development into two large processes (Apel et al. 2013; Pohl et al. 2005; **?**). Domain engineering focuses on

developing the platform (a.k.a., engineering for reuse) with sub-processes, such as scoping the platform in terms of the features to build, or creating and quality-assuring the platform. Application engineering derives concrete products from the platform (a.k.a., engineering with reuse), including eliciting requirements from a customer to identify the required features, developing customer-specific adaptations, and quality-assuring individual products.

From an *organizational perspective*, SPLE requires refining roles and responsibilities. For instance, a platform team is typically established, as well as dedicated feature teams. Especially feature teams work with only a small part of the functionality and gather more focused expertise, which allows them to quickly react to changes and thereby facilitates becoming more agile (Ghanam et al. 2012; Olsson et al. 2012). As such, organizations likely need to revise their structures, roles and responsibilities, as well as the means of collaboration when introducing SPLE.

Despite the benefits (Berger et al. 2020; Krüger and Berger 2020b; van der Linden et al. 2007), and given the upfront investments (Clements and Krueger 2002; Krüger and Berger 2020a; Krüger et al. 2016; Schmid and Verlage 2002), adopting a product line is still a major challenge for organizations. In fact, most organizations start developing a set of similar variants by copying an existing system and adapting that copy, an ad hoc technique called clone & own (Dubinsky et al. 2013; Stănciulescu et al. 2015). This reuse strategy does not scale with an increasing number of variants, and with increasing maintenance costs, many organizations start to re-engineer a product line from the variants and to re-organize their organization, processes, and business strategies. For this purpose, organizations need to assess their current maturity of SPLE, define their improvement goals, and map actions to the identified gaps between the current maturity and the goals. Notably, given the diverse concepts and practices for SPLE (Berger et al. 2020), and the typically large scale of the respective software systems, an organization's maturity usually does not completely pertain to one of the two extremes of ad hoc clone & own and full SPLE with an integrated platform (Antkiewicz et al. 2014), but ranges somewhere between these extremes.

Evaluating the *maturity* (or, *performance*) of SPLE in an organization is challenging. Various measurement techniques have been proposed, ranging from cost and decision models targeting SPLE (Ali et al. 2009; Khurum et al. 2008; Krüger 2016; Thummalapenta et al. 2010) to metrics used to measure the product lines themselves (Berger and Guo 2014; El-Sharkawy et al. 2019; Montagud et al. 2012). A prominent evaluation technique for SPLE is the *Family Evaluation Framework* (FEF), assessing the performance of SPLE (not to be confused with the performance of the product line) in an organization along the four BAPO concerns (van der Linden 2005; van der Linden et al. 2004; Pohl et al. 2005). Each concern is sub-divided into three or four aspects, assessed at one of five levels of maturity. However, as with many of such frameworks, how to apply the framework and analyze its results is far from obvious for an organization. It is also not completely clear where the exact benefits and challenges of applying the framework lie. Organizations need guidance based on substantial and systematically elicited experiences, ideally from tailoring and applying the framework on a range of real-world systems (product lines). In fact, while the FEF comes with a toy case study illustrating parts of its application, to the best of our knowledge, no detailed case study has been presented on applying the framework in the large.

We address this gap by reporting on the tailoring, application, and results of applying the FEF on nine product lines in the aerospace domain (aircraft simulators).

Furthermore, we report experiences in terms of benefits achieved and challenges faced, aimed at supporting industrial practitioners assessing their own product lines, and researchers working on better assessment and benchmarking techniques for product lines or variant-rich software systems in general. Our study relies on action research (Easterbrook et al. 2008), as we worked closely with an industrial partner while designing the study, adapting the methodology to address our research questions, and observing the results.

We address the following two research questions:

RQ₁ *How can the FEF be used to assess large product lines?*
We operationalized the relatively abstract and declarative description of the FEF's maturity levels, each pertaining to one of 13 aspects (classified into the BAPO concerns), and applied our operationalization on nine product lines. Specifically, we investigated four sub-research questions:
  RQ₁.₁ *How can the FEF be operationalized and tailored to a target domain?*
  For instance, what are the concrete steps to identify the relevant product lines, what kinds of preparation are necessary?
  RQ₁.₂ *How can the necessary information be elicited?*
  For instance, are semi-structured interviews sufficient, and which stakeholder roles should be interviewed?
  RQ₁.₃ *How to analyze the elicited information?*
  For instance, what level of transcription is needed, and how can diverging or opposing information about individual aspects be consolidated?
  RQ₁.₄ *How to decide about actions based on the results?*
  For instance, how do diagrams help and what criteria should be taken into account when deciding about steps to improve the maturity level?
RQ₂ *What are the experiences of measuring SPLE maturity?*
We elicit the benefits achieved, and the challenges faced, when applying the FEF. So, we aimed to identify the FEF's actual value for assessing the maturity of SPLE at a large organization.
  RQ₂.₁ *What are the challenges?*
  For instance, what are suitable caveats and what pitfalls may occur when assessing the maturity of SPLE?
  RQ₂.₂ *What are the benefits?*
  For instance, is creating awareness about possible improvements in an organization's SPLE maturity a core benefit?

In our study, we learned that the FEF requires considerable adaptations and knowledge about the assessed product lines. For example, we defined 67 questions, and we involved three stakeholders for each product line in semi-structured interviews. We had to adapt the questions to the domain terminology and explain SPLE concepts during the interviews in order to create a common ground, which was only possible due to the knowledge the interviewer obtained before. While these adaptations required substantial efforts, and while we faced further challenges—for example, to keep the focus during the interviews, to align software engineering concepts or to justify the FEF assessment—the final impression was that the benefits outweigh all the efforts. For instance, all stakeholders became more knowledgeable, we established a common knowledge base, and established comparable measurements. We hope that our insights help other organizations in their efforts to assess

their SPLE practices, and researchers in designing new techniques that extend and facilitate maturity assessments for SPLE.

We proceed by providing an overview of the details needed to understand this article in Section 2. Thereafter, we explain our study design in Section 3. We report our experiences regarding the operationalization and tailoring of the FEF in Section 4, and discuss the challenges and benefits we experienced in Section 5. We describe related work in Section 7, threats to validity in Section 6, and conclude in Section 8.

2 Background

In this section, we describe the background necessary to understand this article. We report on cost models and scoping techniques for software product lines, describe the FEF, and characterize the organization in which we conducted our study.

2.1 Cost Models for SPLE

As several researchers have surveyed, multiple cost models for SPLE have been proposed (Ali et al. 2009; Khurum et al. 2008; Krüger 2016; Thurimella and Padmaja 2014). A cost model aims to support an organization while making an initial assessment whether a product line would pay off or not. To this end, a cost model defines a number of cost factors, which are properties of the project, involved stakeholders or domain that have an impact on the costs (e.g., lines of code, training, tooling). Cost models may be intended for different stakeholders (e.g., management, project lead) and comprise cost factors of the corresponding granularity (i.e., organizational costs, design modifications).

Of the existing cost models for SPLE, SIMPLE (Clements et al. 2005) and COPLIMO (Boehm et al. 2004) may be best known, and illustrate the differences in granularity. SIMPLE is intended to support an organization's management. For this purpose, it defines five so-called abstract cost functions, representing costs for organizational aspects, costs for developing the integrated platform, costs for reusing features, costs for developing unique software parts, and costs for maintaining the product line. However, these functions are not defined further and must be either estimated or filled in based on another cost model. To this end, an organization may rely on COPLIMO, which covers the costs for the platform, reuse, and unique parts. COPLIMO defines more fine-grained cost factors, such as lines of code, design modifications, and number of products. As such, cost models are a helpful means for an organization to assess and monitor investments, pay-offs, and risks, and to plan a product line. Major drawbacks of cost models are the unreliability of cost estimations in software engineering, that cost factors or costs must be estimated at the beginning of a project, and that the pure cost-based perspective may disregard other relevant factors (Boehm 1984; Jørgensen et al. 2009).

2.2 Scoping and Assessment Techniques for SPLE

Several researchers have been concerned with scoping, planning, decision-making, and measuring the maturity of a product line (Ahmed and Capretz 2011; Ahmed
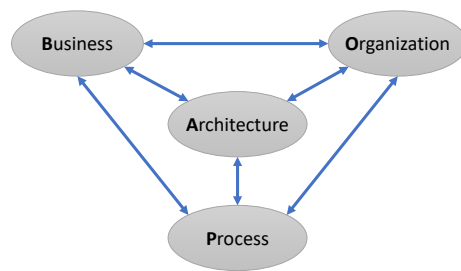
Fig. 1 The BAPO model, illustrating key concerns of SPLE and their inter-relations.

et al. 2007; Bosch 2002; Kalender et al. 2013; Koziolek et al. 2016; Marchezan et al. 2019; Niemelä et al. 2004; Rincón et al. 2018, 2019; Schmid et al. 2005; Tüzün et al. 2015). PuLSE (Bayer et al. 1999) is arguably one of the most detailed of such methods, considering various aspects of planning, adopting, and developing a product line. For instance, in the construction phase, PuLSE-Eco and PuLSE-CDA comprise scoping (i.e., deciding what requirements, and thus products, of a domain shall be integrated) a product line and assessing the costs as well as benefits; while the evolution phase is concerned with monitoring and evolving the product-line infrastructure. Moreover, PuLSE comprises a maturity scale, assessing the maturity of SPLE based on what and how components of PuLSE are used in an organization. This scale includes four levels: initial (components are used independently), full (construction phase is fully implemented), controlled (PulSE is fully implemented and traced), and optimizing (PuLSE is optimized throughout iterations). As we can see, this maturity scale is heavily dependent on PuLSE, which does not allow to assess the maturity of product lines that have been developed without it.

2.3 The Family Evaluation Framework (FEF)

To the best of our knowledge, the FEF is the most comprehensive and generally applicable SPLE benchmarking and assessment framework, originating from industry-academia collaborations (van der Linden et al. 2004; Pohl et al. 2005) in large EU ITEA projects around the millennium. The FEF (van der Linden et al. 2004, 2007) aims at evaluating the maturity of the engineering of software product lines or, more general, of variant-rich software systems. This framework is sub-divided into four general concerns to consider when assessing SPLE of an organization. In Figure 1, we illustrate these four concerns (business, architecture, process, and organization) and their inter-relations, conveying that all concerns influence each other and all need to be considered for effective SPLE.

For the organization we investigated in this study, the relationships between the BAPO concerns are important for the FEF assessment and for defining actions based on the outcomes. Particularly, changes in one of the concerns imply changes in others, for example:

Business ↔ Organization: The high-level decisions on the extent of adoption of a product line and how it is marketed to customers affect the business concern and the organization concern, because the organization's structure is modeled based on both, products and domain knowledge.

Business ↔ Process: The organization's processes change depending on whether the products are marketed as single systems or as a product line. Marketing single systems can mean that processes are not propagating the resulting SPLE artifacts to the whole organization (e.g., feature models), but focus on utilizing the internal benefits of the product line, such as higher quality and faster time-to-market. If the products are marketed as a product line, the whole organization requires the corresponding artifacts, for instance, a feature model is needed for development and marketing alike.

Architecture ↔ Business: The relationship between these two concerns is mostly visible in the time planning and external feature models (if these exist). Features that are externally visible and selectable by customers are usually easily selectable in the architecture, for example, as plugins.

Architecture ↔ Process: These concerns relate mostly to the more detailed parts of the processes. The more detailed and technical a process, the more connected it is to the architecture. An example of this are processes for testing, reviewing, and upholding architectural rules, which are strongly intervening both concerns.

Architecture ↔ Organization: Architecture and organization are related in terms of the responsibility of assets. A product line can be divided into components or assets in several ways, but usually the organization needs to support multiple products simultaneously, which means that developers of an individual product do not have full freedom to create a completely different portioning of the product line. Also, this highly relates to the domain knowledge present in different units of the organization. So, the product-line partitioning, and therefore the architecture, is influenced by the organization and vice versa.

Process ↔ Organization: The process and organization concerns are highly interconnected. The processes are carried out by the organizational units defined, and especially individual roles and responsibilities in the processes have their residence in specific units.

The BAPO model (America et al. 2000; van der Linden 2002) provides the basis for the FEF, whose overall structure we show in Figure 2. The FEF evaluates the maturity of SPLE along four dimensions—the BAPO concerns. Each dimension is further sub-divided into three or four aspects, each of which is assessed on a 5-point scale reflecting the maturity of the aspect. For each aspect and level, the framework lists a set of requirements needed to achieve each level. The levels of the different dimensions are not directly connected to each other, so it is possible to reach different levels on each dimension. However, they are indirectly connected in the sense that some progress is usually necessary on all levels to be able to reach higher levels in a specific dimension (the BAPO principle, cf. Figure 1).

The result of an FEF evaluation is an evaluation profile, consisting of four values, one for each dimension, indicating the respective level as the result of the assessment. As an example, consider the architecture dimension (cf. Figure 2). This dimension assesses the configurable platform of the product line based on the aspects asset reuse level, reference architecture, and variability management. The five levels and corresponding qualifications for this dimension are:

1. Independent Development: There is no reuse for developing new products, instead, they are developed individually and separated from each other.
2. Standardized Infrastructure: On this level, the architecture builds on reusing third-party software and the variability provided in it.
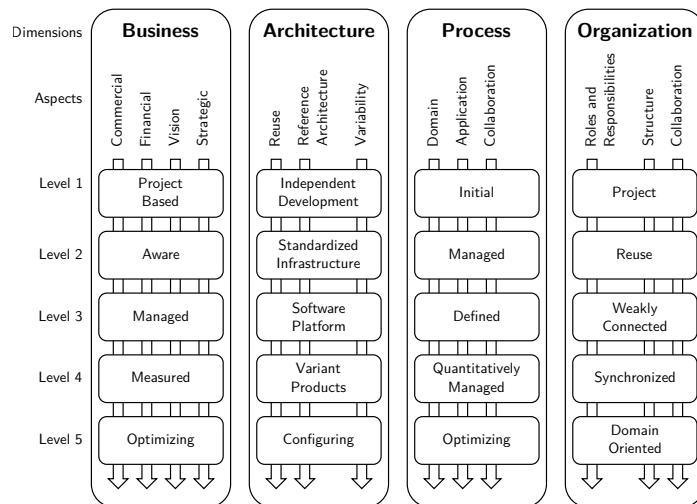
Dimensions	Business	Architecture	Process	Organization
Aspects	Commercial, Financial, Vision, Strategic	Reuse, Reference Architecture, Variability	Domain, Application, Collaboration	Roles and Responsibilities, Structure, Collaboration
Level 1	Project Based	Independent Development	Initial	Project
Level 2	Aware	Standardized Infrastructure	Managed	Reuse
Level 3	Managed	Software Platform	Defined	Weakly Connected
Level 4	Measured	Variant Products	Quantitatively Managed	Synchronized
Level 5	Optimizing	Configuring	Optimizing	Domain Oriented

Fig. 2 The family evaluation framework, adapted from van der Linden et al. (2007).

3. Software Platform: The commonalities of the domain are implemented in a configurable platform, allowing to reuse and combine assets based on a defined reference architecture. However, there is no support for configuration, yet.
4. Variant Products: The organization has a full SPLE reference architecture that specifies the variability of the configurable platform, defines the allowed configurations, and systematically manages the reuse of assets.
5. Configuring: In addition to the previous levels, the integrated platform allows to automatically configure and derive new products, with only marginal differences between domain and application architecture.

While these levels are a helpful means to assess the maturity of SPLE, they also have limitations. First, a higher level in a dimension may not be desired if the current practices are working well for the organization, which is why the results of an FEF assessment must be carefully analyzed before proposing any actions. Second, there are no precise distinctions between the levels, which makes a precise mapping to each level harder. Finally, it is unclear how to best elicit the information needed to assess a level. In this article, we provide extensive experiences on how to tackle these challenges and how to best employ the FEF in practice.

2.4 The Subject Organization and its Product Lines

The Simulation Center. The Saab Aeronautics Simulation Center (hereafter, *Simulation Center*) is an in-house developer of aircraft simulators ranging from prototype simulators, subsystem/system simulators, to training simulators for pilots and ground personnel. Overall, the organization, Saab AB, has around 16 000 employees, and the Simulation Center around 300. The Simulation Center maintains four product lines for external delivery to customers, and 15 product lines for internal "customers" only. While the external product lines consist of a combination of electronics, mechanics, computers, and software, in contrast, half of the internal product lines comprise only software.

SPLE Adoption History and Current State. Adopting SPLE has been a vision at the Simulation Center for many years. As a first step, one of the software-only product lines adopted the basic SPLE principles in 2010 (Andersson 2012), representing an early case of industry adopting SPLE. This particular product line consists of source code in Ada, C, C++, XML, Fortran 77, and Shell scripts, comprising a total of 1.4 million LOC divided into 275 modules, each with its own lifecycle and version management. In addition to the source code, about 20 external libraries are included in the product line. Besides restructuring the software into an actual product line, the Simulation Center also established development handbooks (i.e., development guidelines, documentation templates) and a support organization (i.e., for domain engineering).

After tackling the complexity of variability in this product line, it was obvious for the Simulation Center that using the best practices from the SPLE community was the way towards higher quality, lower development and maintenance costs, as well as shorter time-to-market. Consequently, the management of the Simulation Center decided that all product lines in the organization should adopt these principles wherever the investment costs of the change could be justified. For example, it might not be cost-effective to introduce the overhead of SPLE in a product line with only two variants and almost no new development (Clements and Krueger 2002; Krüger et al. 2016; Schmid and Verlage 2002).

The main goals for switching towards SPLE were:

- To establish a common language among the product stakeholders to describe a product line and variability.
- To unify the processes for product-line-related activities.
- To be able to create fully reusable components.
- To achieve shorter time-to-market and lower development costs.
- To achieve higher quality and only fix each bug once.
- To achieve the Simulation Center's "design once" goal.

In this context, applying the FEF aimed at assessing whether the Simulation Center is on the right track in the adoption of SPLE, defining the goals for advancing each currently existing product line, and providing a more systematic and codified means to assess the performance of SPLE in the organization.

Drivers of Variability. Avionics control systems in general are variability-intensive systems. The Simulation Center's product lines' variability arises from variability in the actual aircraft software (i.e., the diversity of hardware and the different markets in which the aircrafts are sold) as well as from the simulation, since simulation models of different fidelity need to be maintained in parallel. As such, interestingly, the variability is higher in the Simulation Centers' product lines as in the actual aircraft. In addition, secrecy requirements play a role:

- Export control licenses: A particular piece of information (e.g., source code) is not allowed to be used in all places.
- Strict need-to-know policies: An engineer can only access information that they need in the current assignments.
- Secrecy: Information can only be used in approved environments, and usually these environments are not connected to the internet or any other network.

These factors induce additional variability that is completely unrelated to customer features or the drivers above. For example, it might be necessary to develop a particular software component twice: one time with classified data or algorithms, and one time with open data or algorithms.
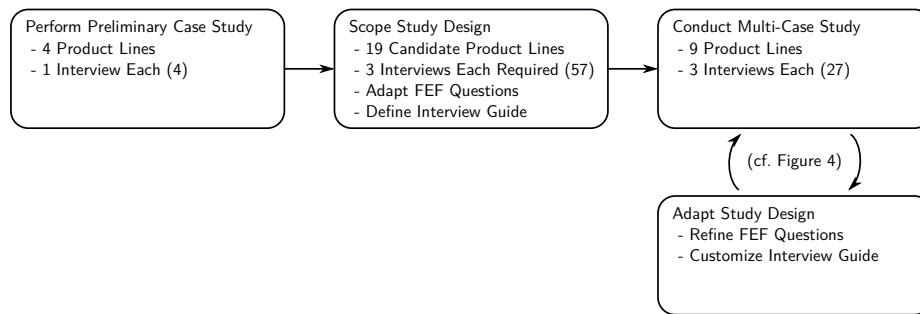
```
┌─────────────────────────────┐    ┌──────────────────────────────────┐    ┌───────────────────────────────┐
│ Perform Preliminary Case Study│   │ Scope Study Design               │    │ Conduct Multi-Case Study      │
│  - 4 Product Lines           │ →  │  - 19 Candidate Product Lines    │ →  │  - 9 Product Lines            │
│  - 1 Interview Each (4)      │    │  - 3 Interviews Each Required (57)│   │  - 3 Interviews Each (27)     │
│                              │    │  - Adapt FEF Questions           │    │                               │
│                              │    │  - Define Interview Guide        │    │                               │
└─────────────────────────────┘    └──────────────────────────────────┘    └───────────────────────────────┘

                                                                                    (cf. Figure 4)

                                                                            ┌───────────────────────────────┐
                                                                            │ Adapt Study Design            │
                                                                            │  - Refine FEF Questions       │
                                                                            │  - Customize Interview Guide  │
                                                                            └───────────────────────────────┘
```

Fig. 3 Overview of the study design.

3 Study Design

In this section, we report our study design, namely what research method we employed at the Simulation Center and on what product lines. We summarize the study design in Figure 3.

3.1 Methodology

At first, we conducted a preliminary case study together with three students at the Simulation Center. This case study was welcomed at the Simulation Center in their endeavor to find metrics and decision tools to support their investment in SPLE methods. During this study, we conducted four interviews for four product lines in order to understand the questions of the FEF. Our insights from this initial experience indicated that the questions required some adaptations and extensions to address actual issues that occur in practice. Moreover, we found that we needed to involve different stakeholders to obtain complementary insights on each of the FEF aspects. Our assessment of these product lines was considered valuable and insightful during a presentation with managers and stakeholders of the product lines. Still, we concluded that a larger study with adapted questions was needed to tackle our research questions, and to derive actionable results for the Simulation Center.

To further instantiate the FEF and gain more detailed insights on its application, we decided to conduct a multi-case study (Yin 2003) based on action research (Easterbrook et al. 2008). A multi-case study builds on several cases, which helps to generalize data and synthesize observations from multiple instead of a single source—improving not only the internal, but also external validity (Siegmund et al. 2015). For this purpose, we assessed the maturity of nine product lines, each involving different stakeholders and system properties. During this multi-case study, we employed interviews to obtain qualitative responses from three stakeholders (i.e., a manager, an engineer, and a technical lead) of each product line. We defined a structured interview guide, and the first author of this article conducted the interviews. He transcribed the responses into a spreadsheet that we used to synthesize our insights.

To cope with differences of the product lines investigated (e.g., pure software versus software combined with hardware), we needed to adapt our interview guide. For instance, we included product-line-specific questions and removed those that are irrelevant for a specific stakeholder role. So, we built on action research by reacting to the

demands and problems that occurred at the Simulation Center. Arguably, this is more helpful to gain practical insights on the real-world application of the FEF compared to simply sticking to the initially defined interview guide. To track the changes we employed, we marked each question with the FEF aspect it is concerned with, relevant stakeholders, and whether it was product-line- or organization-specific. We report the questions of our interview guide, the results, and our experiences in Section 4, Section 5, and Appendix A. When the results included organization-internal information, we translated that information into general statements to avoid disclosures.

3.2 Subject Product Lines

The product lines that we assessed for this study included different components in aircraft development and training simulators. In Table 1, we provide an overview including the type, the team size, and the application domain of each product line, of which several comprise more than 1 million LOC. Some of the product lines in the list are in fact two product lines, representing two different generations of the simulator and its components—a current and a legacy one. In total, there are 19 distinctive product lines at the Simulation Center, nine of which we evaluated according to the FEF. We can assign each product line to a certain type:

– **SW** refers to pure-software product lines.
– **SW, HW** refers to product lines that include software and hardware.
– **SW, HW, Int** refers to product lines that include software, hardware, and the integration of products from other product lines, representing multi product lines (Rosenmüller and Siegmund 2010).
– **Doc** refers to a product line where all assets and products are documentation.

The origins of the different product lines vary greatly. Some are the result of systems evolving over decades, others are newer and more clearly defined. In general, the outer bounds of each product line are clear. What varies most is the degree of separation between the platform development (i.e., domain engineering) and the product development (i.e., application engineering). For existing products, it is clear what the customer-specific features are, but the product-line platform has usually not been systematically scoped. Also, how features are defined varies between product lines, and features are mostly defined in some kind of manifest file. In contrast, feature relations and constraints are never defined in a formal language, but specified only through natural-language documents or are kept in the developers' knowledge. The method of equating features with assets in a manifest file means that features are essentially components, and the actual higher-level concept of features is often missing. Finally, each manifest files comprises between 30 to 300 assets.

  For this study (excluding the initial study with the students), we conducted a total of 27 interviews over a period of 11 months. Approximately, the total effort that has been invested is: 150 hours for the interviews (interviewer plus interviewees), 180 hours for reporting, 100 hours for preparations, and 250 hours for training (including all participants). To complete the assessment of all 19 product lines, we estimated that a total of at least 57 interviews would be needed. However, several of the legacy product lines will probably never be assessed with the FEF. The Simulation Center plans to merge some of the product lines that exist in two versions (i.e., current and legacy architecture). Other product lines will soon reach their end-of-life, and

Table 1 Subject product lines.

Type	Application Domain	Team Size	ID
SW, HW, Int	Development simulators	80	DS
SW	Aircraft simulation	60	AS
SW, HW, Int	Training simulators	30	TS
SW, HW	Computer systems	20	CS
SW, HW	In/out systems	20	IO
SW	Tactical simulation	20	TacS
Doc	Documentation and publications	10	DP
SW	Operating station	10	OS
SW	Visualization	5	VS

therefore no further investments in those will be made. Still, while the Simulation Center will not assess such product lines, the efforts of using the FEF will facilitate working with those. For example, the education and information material for SPLE is used throughout the whole organization.

3.3 Roles and Responsibilities

Multiple stakeholders of the Simulation Center have been involved in this study. In the following, we describe the different roles the stakeholders had and the corresponding responsibilities:

– **Interviewer:** The interviewer, namely the first author of this article, is the lead of the FEF evaluation at the Simulation Center. As such, he was responsible for conducting the interviews, writing internal reports, managing the SPLE training, and communicating to the product-line management.
– **Interviewee, Manager:** The product-line manager is responsible for a number of product lines at the Simulation Center, so these stakeholders may have participated multiple times in the interviews. Each manager is responsible for the long-term development, coordinating between projects, and implementing processes. Each manager invested about 2.5 hours into each interview and 1 hour for preparing as well as prioritizing the other interviews.
– **Interviewee, Technical Lead:** The technical lead defines the technical development of a product line within a defined scope, for example, usually for several projects. They are experts on the product-line architecture and the strategies employed therein. Each technical lead invested about 2.5 hours into their interviews.
– **Interviewee, Engineer:** The (primarily software) engineer represents someone who develops the product line and knows about the daily work, implementation details, as well as the intended design. Again, each engineer spent about 2.5 hours on their own interview.
– **Project Management:** The project management is not directly involved in the FEF assessment, but received the suggested actions that resulted from it. So, it is responsible for prioritizing the actions and defining a development road-map for each product line.
– **Product-Line Management:** The product-line management is the group of all product-line managers of the Simulation Center. They define the overall vision of the SPLE initiative and have asked for the FEF assessment. As a
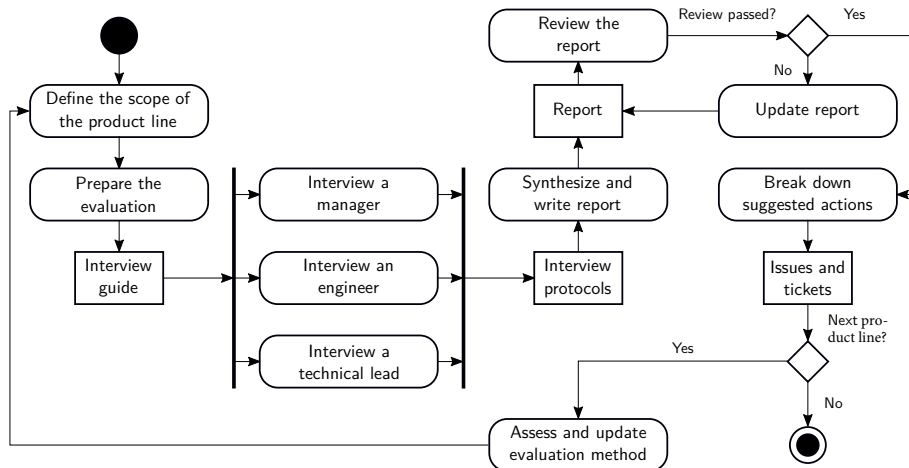
Fig. 4 The workflow of the assessment method.

result, they invested their efforts into the interviews, specific SPLE training for the management (a four-hour workshop), and internal coordination as well as decisions.

We use the same terminology (i.e., particularly for the interviewer and interviewees) throughout the article to refer to the same stakeholders.

4 Tailoring and Applying the Family Evaluation Framework – RQ$_1$

We now report how we applied the FEF in the following steps, illustrated in Figure 4:

1. Define the scope of each product line (Section 4.1).
2. Prepare the evaluation (Section 4.2).
3. Conduct interviews with stakeholders of each product line (Section 4.3).
4. Synthesize interview results and write a report (Section 4.4 and Section 4.5).
5. Review and update the report with the interviewees (Section 4.5).
6. Identify actionable findings and report them to product-line owners (Section 4.6).

Finally, we report how the Simulation Center intends to implement the FEF as a continuous evaluation strategy, as well as we discuss our experiences to answer **RQ$_1$**.

4.1 Define the Scope of each Product Line

In the beginning, we found that it was not clearly defined what the scope of each product line is. To address this problem, we created a template with the following entries to be filled in by the product-line managers:

– **Name:** The name of the actual product line, not to be confused with the family of currently existing product variants.

- **Definition:** A definition of what the product line is about, as a first attempt of scoping it. In our case, the definition helped to think about possible future products, and not only the current products in the product line.
- **Purpose:** A description of the purpose of the product line (i.e., why it exists). This entry helped us to understand the driving factor for creating this product line (e.g., the Simulation Center's "design once" goal, shorter time-to-market).
- **Customers:** A description or list of customers of the product line. Customers provide input to the application-engineering process and receive the outcome of that process (i.e., the tailored product).
- **Current product variants:** A description or a list of the currently existing products. We asked for this clarification to understand what the Simulation Center considers as a product of a product line, for example, with respect to binding time.
- **Features:** List of features, ideally in the form of a feature model. But, it could also be a description of the organization's use of high-level requirements, features, configuration switches or similar information.
- **Organization:** A description of the organization around the product line, such as management, teams or division between domain and application engineering.
- **Architecture:** A high-level description of the product line's architecture with focus on SPLE concepts, for instance, binding time and mode (Berger et al. 2014b), plugins, and configurations.
- **Process:** A description of the processes used for engineering the product line. We focused on SPLE-specific processes, such as finance models for the platform or communication between domain and application engineering.

The first entries helped us to obtain a general understanding of each product line, while the last three entries already align to the BAPO dimensions. We remark that we left out explicit entries about the business dimension (it was included in the process dimension). The reason was that all product lines contribute to a larger system that the Simulation Center sells to its customers. So, most of the product lines do not have their own income connected to sales.

The manager of each product line filled in the template and presented it to all other managers at the Simulation Center for feedback and dissemination. These presentations were part of a large workshop with all managers, moderated by the first author of this article. All managers are a team below the second-level manager (responsible for the Simulation Center), so they were well aware of their respective product lines. Each manager presented their product line(s) to the others within 15 minutes. During the presentation, the answers in the template were shown on a projector. After the presentation, all other managers had the opportunity to provide feedback. The feedback was documented by the presenting manager, and resulted in updates of the template.

During this workshop, it became clear that some product lines were uncontroversial, whereas others required discussions to be established. The most common cause of discussions were different views on the scope of a product line, for example:

- Some product lines were considered to represent two separate product lines with common assets between them. Both representations could be correct, and the decisive factors for the final choice were usually not found in the technical area, but in the organization or business dimension.
- The presented product line was too broad in scope, also covering other product lines. This was usually the case with product lines that used the variants from

other product lines as assets (i.e., DS and TS in Table 1). Such product lines should only integrate the products from other products lines into an external product, but usually the developers also felt responsible for the internal behavior of their features, and thus included other product lines in the scope of their product line.

In the end, we obtained an understanding for each product line and had a brief summary of its core characteristics. This information helped us to define the questions for our interviews and to include the questions relevant for each product line.

4.2 Preparations for the Interviews

To prepare further for the interviews, the interviewer collected additional information on each product line. For this purpose, he read available documents, such as system descriptions, requirements documentation, formal processes, and informal way-of-working descriptions. The knowledge the interviewer collected helped to guide the interviewees and support them in understanding the questions defined. For example, not all interviewees knew the terms domain engineering and application engineering, while their concepts were clear. By enriching his knowledge, the first interviewer was able to map the research terminology to the organization's terminology, for instance, referring to different teams or roles that are involved in developing the product line.

Afterwards, we compiled the questions for our interviews (cf. Appendix A). In total, we designed 67 questions, comprising 47 for managers, 42 for technical leads, and 38 for engineers. So, we adopted our interview guide according to the different roles of our interviewees and their respective knowledge in the BAPO dimensions, but did use the same questions for all product lines. To this end, we built on the BAPO dimensions and example questions described by van der Linden et al. (2007). We converted the example questions to be understandable by stakeholders who do not have SPLE knowledge, and customized them to the domain terminology of the Simulation Center. For example, we converted "software asset" into "configuration item." At the Simulation Center, a configuration item is defined as a software module with an own life-cycle, versions, requirements, test cases, and other artifacts; essentially representing an SPLE asset, the implementation of a feature. Using the terminology established at the Simulation Center facilitated communication and kept the focus during the interviews on the actual FEF assessment.

Customizing the questions to the organization's needs is an investment that we believe pays back in two ways:

1. The *interviewer* improves their knowledge about the FEF methodology and must contemplate on the practical meaning of the different concerns in the BAPO model for their organization. Essentially, the interviewer must always answer the following question while customizing the FEF questions: *What does this mean for us?* Missing this understanding and using the generic template would impair the conduct of the FEF assessment, potentially rendering the results useless.
2. The *interviewees* receive the questions in a familiar terminology, language, and aligned to the organization's practices. So, they can focus on answering the questions, instead of trying to understand, translate, and match generic questions.

In this particular study, the same organization manages 19 product lines in the same domain. As a result, the investment of customizing the questions is distributed over

all these product lines and over time, since the FEF assessment will be repeated several times for each product line in the future.

Converting and customizing the interview questions means that they are less directly mapped to the actual BAPO levels, which is two-fold. On the one hand, the interviewer requires a deeper understanding of the product line that is assessed. While the interviewer in our study is involved in the Simulation Center and improved his knowledge by reading additional documentation, most knowledge can only be elicited during the actual interviews. So, the simplest answer to a question will not reveal the BAPO level, which is why it is important to encourage interviewees to elaborate on their answers (e.g., by asking follow-up questions). On the other hand, the missing connection to BAPO levels improves the understandability of the questions and can provide more honest answers, as the interviewee does not need to worry about the BAPO level their answers may indicate. Moreover, to obtain as much reliable information as possible, we started each interview by introducing the BAPO levels and explaining that these are neither grades nor that higher levels are necessarily better. Each level may be a perfect fit for the specific product line and the investments to reach a higher level may simply not be worth the returned value. Finally, the conversion of questions has one more trade-off: The analysis in terms of BAPO levels must be done by the interviewer, which results in a more consistent synthesis, but may bias the assessment due to missing knowledge. Consequently, there will always be a difference in the interpretation of the BAPO levels.

One problem of the FEF is that many dimensions and their questions are rather abstract. For example, the levels of the architecture dimension are open to different interpretations, depending on the specific architecture used in the product line that is assessed (cf. Section 2.3). To overcome this problem and simplify the analysis, we compiled 14 questions to assess this dimension. We asked most of these questions to technical leads and engineers, who have a detailed understanding of their product-lines' architectures. Overall, we designed 11 categories of questions to make the FEF assessment more understandable in this particular domain and facilitate the analysis of results:

- 7 general questions about the interviewee, including their experience and subjective view on the product line.
- 4 general, high-level questions on the business dimension.
- 3 business questions on the SPLE vision and goals of the product line.
- 4 business questions regarding the SPLE planning, such as separation of plans for domain engineering and application engineering or business value.
- 14 technical questions on the architecture of the product line.
- 1 general question on the processes related to SPLE.
- 12 questions related to domain-engineering processes.
- 4 questions related to application-engineering processes.
- 9 questions on collaborative processes.
- 8 questions related to the organization of the product line and projects.
- 1 open question where the interviewee could add anything they like or give feedback on the interview itself.

Using these questions (cf. Appendix A for a more detailed list) to open the discussions on each subject with the interviewees, we aimed to obtain detailed insights into a product line to assess its SPLE maturity based on the FEF.

4.3 Conduct Interviews

Early on, we made the following decisions:

– Each product line should be evaluated by conducting three interviews with:
  – the closest manager of the product line;
  – the highest technical lead of the product line;
  – an engineer working on the domain engineering of the product line.
– Each evaluation should result in an extensive report and a presentation summary.

We decided to pick these interviewees because most questions in the FEF can be answered by a single expert (i.e., the technical lead), but this would result in an incomplete picture of the actual situation. Instead, the three roles provide different perspectives on the product line, resulting in a composite picture based not only on the answers themselves, but also the differences between the same questions, which can be a valuable addition. For example, in some cases the manager pointed to a defined process for a specific activity. However, the other two interviewees were not aware of that process. Afterwards, we concluded that there was a defined process, but it was not communicated. We could only derive this finding because we interviewed multiple stakeholders and used the same questions.

The decision to write reports was defined by considering the situation of the Simulation Center. It was clear that, for example, the results of the FEF assessment, suggested improvements, and change management, would be long-term processes, potentially spanning several years or decades. The Simulation Center cannot ensure that the same personnel would be available for the whole life-span of the product lines or change projects. So, it decided to document everything as thoroughly as possible to not lose valuable information.

The 19 product lines at the Simulation Center would require 57 interviews based on our methodology. Besides skipping those that will be potentially merged (cf. Section 3.2), we conducted a prioritization by assessing for which product lines the FEF evaluation was estimated to provide the most value. As we did only initial student projects with informal results and had no key performance indicators for the maturity of SPLE, the prioritization was based on:

– The *current* and *estimated future* number of products in the product line. A higher number of products yielded a higher priority as very few products can be handled without the extra overhead of SPLE.
– The *current level of development* (improvements and bug fixes in a time frame). A high level of development yielded a higher priority as the benefits of improvements are only visible if changes are made and new products are produced.
– The *size of the organization* maintaining and developing the product line. A larger organization yielded a higher priority as improvements affect more people.

For example, a product line with two or three products, no new products planned, only maintenance and bug fixes, as well as very little new development was of low priority. In contrast, product lines with a higher number of products, a complex variability structure, and a larger organization were considered to gain more from improvements based on the FEF assessment, therefore obtaining a higher priority.

After prioritizing the product lines and inviting interviewees of each respective role, we started with the actual interviews. Each interview began with an introduction of roughly 30 minutes into the basic concepts of SPLE. Then, we asked the questions

prepared for each role and provided the opportunity for an open conversation at the end to allow the interviewee to add any comment they liked. Overall, each interview session took between 1.5 and 2.5 hours.

While some questions required only short answers, others could raise further discussion between interviewee and interviewer. The interviewer took notes during the interview to document the final answers to each question. In parallel, each interviewee could read the notes on a large screen and comment on them. We documented the discussions raised for some questions solely if they were important for the FEF assessment. For instance, if the discussion raised a previously unanswered question, we recorded it instead of noting no answer.

For all but one of the product lines, the roles of manager, technical lead, and engineer turned out to be a good distribution to assess the maturity. However, in one case this did not fit. This product line does not have the same balance between product-line management, project management, and technical leadership as the others. Instead, for this product line, the project itself owns each decision. During the interviews, this became apparent as both, the manager and the technical lead, often referred to project management. Although this was a special case, involving the project management into the FEF evaluation would probably have given a more comprehensive and complete picture of the SPLE maturity.

4.4 Analysis of the Interviews

Initially, we audio recorded interviews and transcribed them later on. However, we did not record most of the interviews. Instead, the interviewer took notes for each question, which he showed to the interviewee simultaneously for reviewing. The notes included the final answers to each question as well as additional comments and insights that were relevant for the FEF assessment. We changed the procedure from recording to taking notes since some interviewees where uncomfortable with recordings, mostly because taking notes enabled the interviewee to directly review the interpretations done by the interviewer.

We used the documented answers, the interviewer's memory, and the previously obtained knowledge of the product lines and organization to judge the level of each BAPO dimension. During this phase, we experienced that a single answer was seldom precise enough to actually judge such a level. So, we synthesized the judgment from all answers and the discussions that we documented. In most cases, this resulted in a clear judgment for each level when compared against the requirements defined.

If any answer was unclear, we briefly contacted our interviewees to ask for additional information. We remark that we did not attempt to form a consensus between the interviewees in cases when they provided different answers. In our experience, finding discrepancies between the answers was key for the overall assessment of SPLE maturity.

Example of analysis. An example of a question that was discussed is: *How is the connection between a variant and its assets managed? Do you ever write the names of the variant on the assets itself?* This is a vital question for SPLE: If the processes or architecture require that each asset itself is directly linked to a specific product variant, the platform does not resemble an actual product line. For one of the product lines, the answers were:

- `Manager:`
  "The environment where the asset is allowed to be installed in is written on the asset itself, this in turn makes a weak connection to the available variants (as a variant can only exist in one environment usually). Very few assets are in other ways connected to a specific variant."
- `Technical Lead:`
  "The strategy is to never write on the asset where it will be used. But there are exceptions."
- `Engineer:`
  "We never write on the asset where it is to be used. However, a single product variant is often the driver for the development of an asset or feature."

The answers to this question indicate no clear consensus on this topic, which in itself is an important finding. All interviewees seem to be aware that "writing the variant name on the asset" is bad, but it is unclear to what extent this is still done.

This was one of 14 questions in the architecture dimension of the BAPO model. The product line was assessed to be on architecture level 2 (see Figure 5), *Standardized Infrastructure*, with the reasoning:

- There are techniques to reuse assets. Mostly, the product line is based on late binding of assets that are used, for example, with configuration files that are parsed at runtime. A general strategy for reuse does not exist, but in practice it works well with its current "ad hoc" strategy.
- There are three separate architectures in this product line. The question is whether the scope of this product line is too wide, and should be divided into three.
- The management of the variability of each asset is done within each asset. There is a layer of integration between the assets, mostly belonging to the application engineering, not an actual platform.

To evolve into level 3, *Software Platform*, the following was suggested as a start (not a complete list of actions):

- Describe the architecture further. Contemplate why some stakeholders consider three separate architectures to exist.
- Investigate if it is possible to standardize the interconnection of assets.
- Investigate if it is possible to standardize the variability management. Currently, it is scattered over source code, build processes, and configuration files, as well as some manual moments during installation.
- Investigate if the concept of features can be used instead of making markings on the assets with names from application engineering (such as product names).

We can see from this example that aligning the answers of different stakeholders with each other and the BAPO levels can be problematic, due to the vague distinctions between levels (cf. Section 2.3). Consequently, our real-world experiences are highly valuable for practitioners who intend to employ the FEF themselves.

4.5 Synthesis of Results and Reporting

With the interview results as basis, we wrote a report for each product line. We divided the contents of the report according to the four BAPO dimensions. Each of these sections started with a summary of all corresponding interview answers,
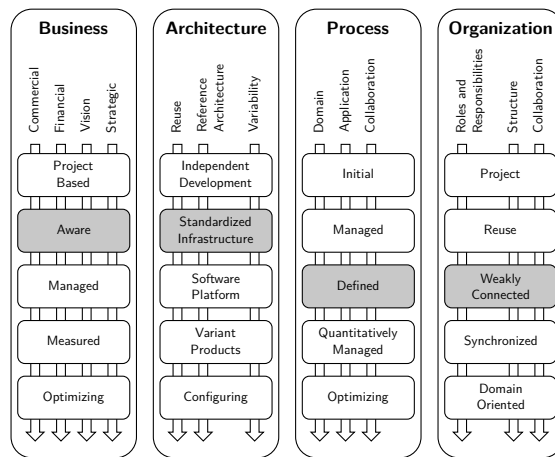
Fig. 5 An example assessment of one product line according to the FEF.

followed by subsections of the FEF assessment according to the BAPO levels, and suggested actions for reaching the next BAPO level. We evaluated the reports in a meeting with all interviewees. These evaluations did not result in changes of the report, but comments in a separate document. It was decided to not change the report at this point, as even if it contained some faults and misunderstandings, it reflected the conception of the interviewer at the time of the interview. In practice, we identified no major faults in the reports, and the comments from the evaluations where mostly additional information. Some evaluations resulted in discussions between the interviewees, indicating different views on whether an established process existed or not. The report ended with a diagram, as the one we show in Figure 5, indicating for each dimension which level the product line was assessed on.

Additional interviews after publication. After we published the reports for each product line internally and disseminated them at the Simulation Center, feedback from readers came in. Everyone who had feedback agreed with the reports, and sometimes had additional remarks and information. Two stakeholders requested to meet, so that they could give their answers to the questions used in the interview. This resulted in an addendum to the original report, however the addendum was only spread to the managers of the organization.

4.6 Break Down the Suggested Actions into Issues

A report itself will not automatically change anything in the organization. To establish the suggested changes, the Simulation Center held a series of meetings and workshops, with the following goals:

– Break down each suggested action into smaller Scrum or Kanban stories.
– Find financing for each of the actions. In some cases, the affected project could rely on its current budget, but more often external financing from central functions of the organization was needed.
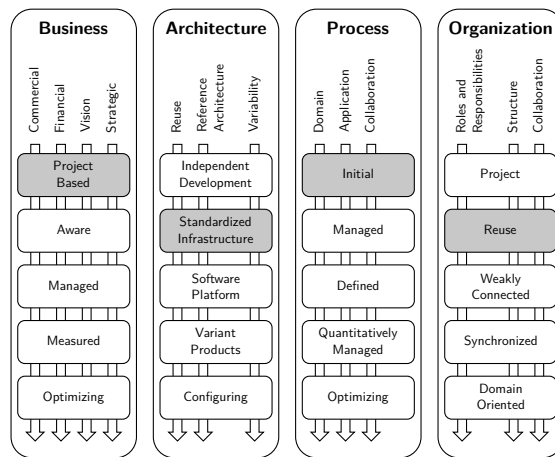– Prioritize each action.

Fig. 6 The BAPO diagram for the FEF assessment of Examples 1 and 4.

    – Set the goals of the product line in terms of FEF levels. This did rule out some suggested actions, as these actions aspire for a higher level than the goal.

In the end, the workshops resulted in a number of actions that the participants agreed on for each product line to achieve a specified goal with respect to SPLE.

4.7 Repeat

To establish the FEF assessment as a natural part of the regular work, repetitions of the assessments and their evaluation are necessary. The Simulation Center decided that one year would be good interval, at least for the first rounds of the FEF assessments for each product line. So, the next round of assessments will be an opportunity to have a retrospect on the current assessments, reports, and changes that we describe in this article.

4.8 Selected Results from the Measurements

In the following, we exemplify results for the product-line assessments. For confidentiality reasons, we cannot disclose the names of the product lines.

Example 1 (Figure 6): This diagram is the result for one of the more complex product lines with the largest amount of legacy source code, documentation, and defined processes. Also, it represents a product line where all variants are developed in a single long-term project. We obtained low levels in the BAPO dimensions, but the FEF measurement also showed that there are higher ambitions in this product line. Even during the FEF assessment, the Simulation Center proceeded to establish a platform from which new products can be built. So, there were ongoing changes in the product line to make teams more focused on either domain or application engineering. Higher levels in the architecture, process, and organization dimensions can be expected in the next measurement of this product line.
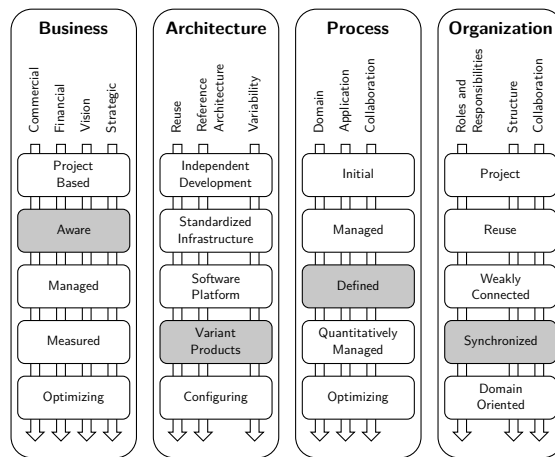
Fig. 7 The BAPO diagram for the FEF assessment of Example 2.

Example 2 (Figure 7): This diagram is the result for one of the more mature product lines, for which SPLE principles were considered from the beginning. So, the product line has a high level in the architecture dimensions, which we expected since this dimension has been the main focus for several years. The organization plans to further improve in this dimension, but the goal will probably never be to reach level 5. Moreover, the product line has a well established set of processes and handbooks, both used internally in the product line, but also by the users of the variants. In accordance with SPLE principles, domain and application engineering are separated.

Example 3 (Figure 8): This diagram is the result for a product line that stands out from the others in the sense that it has very few products, but nevertheless has adopted many of the concepts from SPLE. The product line has only one product that is delivered to all customers and its variabilities are introduced by the customer after delivery. Arguably, this does not resemble a product line, as it comprises only one product. However, we argue that the principles of SPLE are the same, no matter where the variability is introduced. The product line has an architecture that supports this strategy and a well established set of processes and handbooks for both, internal and customer use. Organization-wise, there is only one team, which makes it challenging to assess the level in this dimension correctly.

Example 4: (Figure 6): The diagram for this product line is identical to the one in our first example. For this product line, the Simulation Center spent a lot of effort after the FEF assessment. This product line consists of software, COTS hardware, and in-house developed hardware with variability at different binding times (i.e., early binding and late binding). The teams working on this product line stated that the FEF assessment kickstarted improvements they had on their minds for a long time, but never formalized. Such improvements consist of an SPLE architecture that will probably result in level 3, software platform, and formalized processes. While this product line employs various well-defined processes, these are unfortunately sparsely documented. We expect the next assessment of this product line to have higher levels in the architecture and process dimensions, perhaps also in Business.
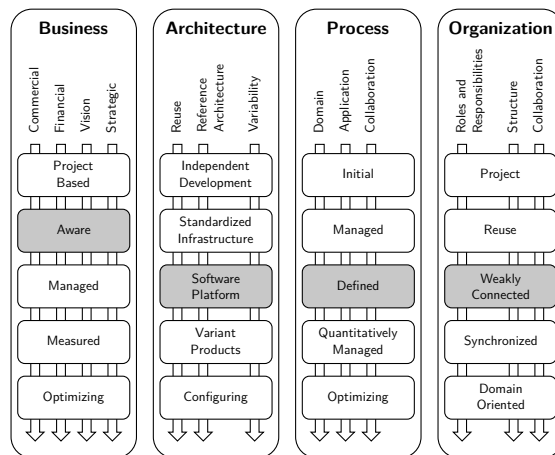
Fig. 8 The BAPO diagram for the FEF assessment of Example 3.

4.9 Expected Versus Obtained Results

Comparing the expected and actual results of the FEF assessment in informal meetings with the technical management of the Simulation Center revealed the following insights:

– Most product lines had a clear and mostly correct perception of their performance in the architecture dimension. The ambitions as well as shortcomings in this dimension were well known in the organization.
– Most product lines overestimated their performance in the process dimension. The level of tacit knowledge compared to formally defined processes and method descriptions was higher than expected.
– The management of one product line with levels of 1 and 2 in all dimensions questioned the FEF assessment with the motivation that "we always deliver[...] on time." After reasoning about the results, the management accepted the assessment with the mindset that a met time plan can also indicate that the plan was too generous, that corners had been cut or that technical debt had been acquired.
– The management of one product line had a clear perception of their abilities in SPLE beforehand, and it showed to be quite correct in the FEF assessment with high levels in several dimensions. They were not surprised themselves, however the rest of the organization had a much lower confidence in this particular product line before the assessment. In this case, the FEF assessment justified the efforts that were put in developing this product line and communicated this to the organization.

These insights indicate that there can be varying perceptions of the goals that are defined in an organization, and that the FEF may help to identify mismatches in the perception and actual goals or the current state. Moreover, the FEF assessment can be a helpful means to communicate and reason about efforts that have been put into developing a product line.

4.10 Summary

Operationalization – RQ$_{1.1}$. We experienced that the FEF is suitable for a regular assessment of large product lines. Depending on the knowledge of the personnel and the adoption level of standard terminology, the FEF must be tailored towards the analyzed domain. To lower the threshold for the organization and facilitate the comprehensibility of the assessment, we found it valuable to invest into preparations before conducting interviews. Moreover, it was highly valuable, and actually necessary, to gain expertise on SPLE, the FEF, and the product lines to explain the process to the interviewees.

RQ$_{1.1}$: Operationalization

For the operationalization and tailoring, we experienced that it was helpful to:
  – *Understand and define the scope of each product line before the actual assessment.*
  – *Adapt the questions of the FEF towards the organization's needs and domain.*
  – *Unify terminology and explain the concepts to the stakeholders.*
  – *Involve different stakeholders into the assessment of a product line.*

Information Elicitation – RQ$_{1.2}$. We considered it valuable to elicit information based on interviews with stakeholders in different roles. The varying perspectives each stakeholder can provide add value to the outcome. Moreover, interviews were the preferred way of eliciting information for the organization, introducing less disruptions. While we relied on some documentation, a large-scale analysis of all available documents seems unreasonable and too expensive.

RQ$_{1.2}$: Information Elicitation

To elicit information, we found it valuable to:
  – *Conduct semi-structured interviews with different stakeholders.*
  – *Assign questions based on each interviewee's role and knowledge.*
  – *Document the responses and make them visible to the interviewee.*

Information Analysis – RQ$_{1.3}$. The thorough knowledge the interviewer collected on each product line proved to be highly valuable, not only to scope the questions of the FEF and facilitate the interviews, but particularly to understand and synthesize data from the documentation. Otherwise, a precise mapping of the responses to the FEF levels seems not possible, unless the levels should be determined in a self-assessment. Moreover, this knowledge is helpful to synthesize different answers, particularly if they comprised discrepancies.

RQ$_{1.3}$: Information Analysis

While analyzing the available information, it was important to:
  – *Synthesize information from all interviews to assess FEF levels.*
  – *Consider discrepancies between answers in particular.*
  – *Evaluate and update the results together with the stakeholders.*

Actions – RQ$_{1.4}$. In this article, we reported our first experiences of using the FEF to assess product lines, but we have not used it long enough to derive conclusions on how to implement actions. At this point, we argue that the levels of the FEF are too coarse and too different to allow to define goals solely based on this assessment. However, we could propose several actions to directly improve the SPLE practices

at the Simulation Center, and the stakeholders of one product line in particular were eager to implement them even without defined goals or a road-map. They experienced immediate benefits in their daily work, arguing that the investment paid off after a short time in a more effective way of working.

RQ$_{1.4}$: Actions

To decide on actions based on the FEF assessment, we considered it valuable to:
- *Derive actions with the stakeholders and break them down into smaller stories.*
- *Prioritize actions according to responsibilities and goals.*
- *Provide feedback on the performance of implemented actions.*

---

5 Benefits and Challenges of the FEF Assessment – RQ$_2$

In this section, we answer **RQ$_2$** by providing an overview of the challenges (**RQ$_{2.1}$**) and benefits (**RQ$_{2.2}$**) of applying the FEF and communicating the results in the Simulation Center. We remark that several of the challenges and benefits are inter-related, for example, the challenge of justifying investments can immediately benefit the challenge of handling change requests. Still, both challenges indicate different origins and potential benefits of addressing them: Justifying investments is concerned with the business dimension and helps to convince stakeholders, while handling change requests is concerned with the process dimension and can help to improve processes as well as organizational structures.

5.1 Challenges

Distinguishing Domain and Application Engineering. We experienced that it was challenging for interviewees to actually describe the benefits of SPLE for the organization and to specify the activities that exist outside of their respective projects. In particular, long-lasting projects tended to embrace more and more domain-engineering activities over time, even though, these should exist outside of the project. This clearly highlights a missing separation and awareness for domain and application engineering, and this discussion can even seem irrelevant to the interviewees as "everything is done in the project anyway." So, the first challenge of applying the FEF is to introduce the right terminology and communicate the benefits and needs for separating application and domain engineering. If this cannot be achieved, the assessment may be doomed to fail, as it may not reveal helpful insights or may be simply ignored.

Focusing on the FEF Assessment. Our interviewees were sometimes keen to lift other problems, unrelated to the FEF assessment. In some cases, it became challenging to keep the interview on track and focus again. Nonetheless, it was helpful and important to not interrupt the interviewees too much, keeping them motivated and potentially identifying relevant information for the FEF assessment. It was not always obvious that a topic was unrelated to the FEF assessment until we dug deeper during the discussion. If we experienced that a topic was not related to the FEF assessment, we recorded it and propagated it to the right forums. Such topics were, for instance,

issues and suggestions on development environments; issues with the incoming and outgoing delivery routines; and discussions on larger, strategic choices of technology and methods. The challenge is to allocate an appropriate period of time for each interview, plan its recording, and find the right balance while guiding the interview.

Aligning Software Engineering Practices. The Simulation Center employs agile methods, such as Scrum and Kanban, which were sometimes argued to oppose SPLE. Particularly, SPLE and the FEF assessment require a more formal style of working, especially in the interface of domain and application engineering. For example, Scrum assumes a high level of volatility in the requirements, while SPLE usually demands more firm requirements that do not change—but may vary for different products. While there is substantial research on how to combine SPLE with agile methods (Klünder et al. 2018; da Silva et al. 2011), some details still seem unclear. The challenge is to identify how to best incorporate useful practices that are established in an organization with SPLE, and communicating that this can be achieved.

Identifying Problems Outside the Product Line. A particular problem with a product-line architecture is its misuse, for example, by products that modify it without considering its dependencies to other products. At the Simulation Center, task forces are an established concept to solve a specific problem, but they do not take other products, future products or maintainability into account. Their goal is to solve the problem as fast as possible, usually driven by an urgent deadline—therefore disregarding the boundaries of projects, teams, and processes. While task forces are immensely helpful to resolve problems, they easily create unintended dependencies between assets. Similar to aligning agile methods, task forces with their informal structures are a contrast to the well-defined SPLE practices. The challenge in this regard is to find a solution on how to coordinate task forces with SPLE practices or how to separate them from the product line if needed (e.g., using clone & own to create independent products (Dubinsky et al. 2013; Stănciulescu et al. 2015)). Concerning the application of the FEF, it can be challenging to identify such established practices like task forces that have considerable impact on SPLE, but may not be known by the interviewees, as they act independently of established organizational structures.

Justifying Investments. Another challenge for applying the FEF and for defining recommendations based on the results is to balance the return on investments. Applying the FEF as well as implementing changes does cost, and a justification for these investments can be hard, requiring justification that value is added and why something should be changed that is working. Especially for product lines with few products and for long-lasting projects this can be problematic. If incorporated, but not necessarily documented, organizational structures and tacit knowledge work well without formal processes, any change is hard to justify, as "everything works now." This mindset can be important to critically analyze the FEF assessment, but it may also block important changes. Consequently, a reasonable justification of all investments is needed to convince the involved stakeholders that the assessment or a proposed change will provide value for the organization.

Managing Change Requests. Change requests, representing the suggested actions that originated from the reports, can be hard to manage if they do not represent a specific

customer request or have no immediate return on investments. Most change-request processes are based on projects, which prioritize, plan, finance, and execute each of the requests relevant to them. However, many actions resulting from the FEF assessment are associated with domain engineering, which is usually out of scope for a single project. Without organizational changes, the projects can either accept a change request within their current budget or special financing and resource planning must be conducted. So, an organization may be challenged to initiate proposed changes without first adopting its organizational structure or providing a separate budget, due to legacy structures, processes, and responsibilities. For applying the FEF, the resulting challenge is to identify the right stakeholders to which change requests must be propagated, especially if the responsibilities are not clear.

RQ$_{2.1}$: Challenges

For applying the FEF, we experienced that it can be challenging:
- *To distinguish between domain and application engineering, which may be tangled.*
- *To focus on the FEF assessment during interviews, due to other legacy problems.*
- *To align other software engineering practices with SPLE to convince interviewees that SPLE can be successful.*
- *To identify problems that occur outside of—but affect—the product line.*
- *To justify why the FEF assessment and the proposed actions should be applied.*
- *To manage change requests by propagating them to the responsible stakeholders.*

5.2 Benefits

Disseminating SPLE Knowledge. As a first experience, we found that applying the FEF to assess the maturity of product lines helped to disseminate information, and thus establish knowledge, on SPLE. In particular, assessing the different dimensions broadened the view of our interviewees, who often focused on the architectural dimension of reusing components. This limited the Simulation Center's ability to advance further towards an SPLE organization. To overcome this problem, we focused on establishing a factory metaphor, which made it more obvious for most interviewees that a successful product line also requires standardized interfaces, tools, clear roles, well-defined processes, and a supporting business model. We consider the dissemination of such knowledge as a major benefit, because it improves expertise, establishes a common knowledge base, and facilitates communication.

Connecting Stakeholders. Reviewing the reports that were based on different interviews connected the managers, engineers, and technical leads of each product line with a more unified perspective and established new communication. Previously, they mostly communicated on an issue-by-issue basis or in forums with a specific focus. During the FEF assessments, each interviewee could raise any issue, from small technical problems to the strategic orientation of the product line. The FEF reports provided a basis for discussions that did not exist before, and we consider this to be a major benefit.

Identifying Possible Shortcomings in BAPO Dimensions. Arguably the main purpose of applying the FEF is to assess the current maturity of a product line and identify

opportunities for improvements. In our experience, especially the interviews proved to be a suitable way to collect information and identify potential shortcomings. The main benefit of using the FEF and BAPO are their structure according to different dimensions. This structure provided a defined setting and forum in which we could manage and document shortcomings that the organization were unaware about or that were unclear (e.g., who would be responsible, why did a problem occur).

Defining Road-Maps for Product Lines. The reports we created during our analysis formed a first road-map for the domain engineering of the product lines that lacked clear plans. In particular, several of the product lines we studied had no activities defined for domain engineering. While these activities were performed, they were not distinguished from application engineering, and thus performed in the same process. The FEF assessments showed the benefits of having a separate domain-engineering road-map, more detailed plans, a separate finance model, and separate organizations. Overall, the benefits of separating domain engineering were connected to long-term thinking of technology advances, to the will of keeping the systems up to date with state-of-the-art, and to use the latest development methods and tools.

Setting Goals for Product Lines. The BAPO dimensions and levels defined a concrete and new way for setting goals for a product line. Before, somewhat standardized goals (e.g., technological level, costs, organization size or time plans) existed. However, these goals could not be compared between different product lines. Using the BAPO model in combination with FEF provides a common ground based on which goals can be discussed within and between product lines. This also has the benefit that the stakeholders of different product lines can communicate and discuss more easily about the goals of each product line.

Lowering Maintenance and Development Costs. Higher performance (as measured by the FEF) means lower maintenance and development costs, especially for product lines with a larger number of products and more variability. The first formal product line at the Simulation Center was established in 2010. Since then, the data shows at least a 50 % reduction of maintenance and development costs. Also, the pricing for new products is more precise compared to before. Applying the FEF can raise the awareness for such benefits and result in even more savings by improving the organization's practices even further.

RQ$_{2.2}$: Benefits

Applying the FEF had the benefits of:
- *Disseminating SPLE knowledge among the stakeholders of a product line.*
- *Connecting stakeholders by providing a unified report as knowledge base.*
- *Identifying shortcoming in the BAPO dimensions based on a structured assessment.*
- *Defining road-maps for missing domain-engineering activities of a product line.*
- *Setting goals for product lines that are comparable.*
- *Lowering maintenance and development costs based on the proposed actions.*

5.3 Summary

All product lines in this multi-case study have been developed for more than 10 years. So, they all have a corresponding history with several legacy artifacts in all BAPO dimensions (e.g., established processes, organizational structures, and architectures). The challenges of applying the FEF were mostly connected to bridging the gap between this legacy and current SPLE practices, and coordinating the required changes. However, the FEF provides a standardized assessment that resulted in immense benefits for the Simulation Center. In particular, the distribution of knowledge, alignment of different stakeholders, and ability to define comparable goals proved to be major benefits of the assessment.

Overall, the consensus in the management of the Simulation Center is clear: The benefits of SPLE and applying the FEF outweigh the challenges. In the technical management, the stance is similar, with the addition that the added technical risks that come with an increased level of reuse must be addressed. All stakeholders welcomed the increased focus on domain-engineering activities with the long-term planing that it entails.

6 Threats to Validity

In this section, we describe the threats to validity of our study, using the categorizations of Yin (2003) and Wohlin et al. (2012) as guideline.

6.1 Construct Validity

As we described before, not all stakeholders that were involved in the interviews have been familiar with SPLE concepts. To mitigate this threat, we mapped and unified the domain terminology of the Simulation Center with the terminology of the FEF. Moreover, we explained the SPLE concepts during the interviews to ensure comparable knowledge and a common understanding between the interviewees. To further improve the reliability of our data, we conducted multiple workshops and reviews of the reports we created, allowing all stakeholders to provide additional feedback on the results. So, while we cannot completely prevent this threat, we have a high confidence that our data is reliable, and that our synthesis of the results and actions was reasonable.

6.2 Internal Validity

The internal validity of our study may be threatened due to the product lines, stakeholders, and organization-specific adaptations to the FEF. While we could not completely resolve these threats, we aimed to reduce their potential impact. First, we performed cross-case analyses, synthesizing our results from all nine cases we considered in this article. Second, all stakeholders we interviewed are experts of their respective product line, and we invited stakeholders from different roles to complement their responses. Finally, our adaptations of the FEF may have influenced the outcome of our study. However, our intention was to operationalize the FEF, and therefore the adaptations are one of our contributions.

6.3 External Validity

We considered nine cases for our study to improve the external validity of our results. However, we investigated only one organization, which threatens the generalizability of our results to other organizations. As there exist no experiences on how to apply the FEF in practice, our multi-case study is still a valuable contribution. We are the first to describe how to tailor and use the FEF in practice, and particularly for large-scale product lines.

Another threat to the external validity are the organization's characteristics. First, the Simulation Center must enforce requirements that are not necessary in other domains (cf. Section 2.4). As this lies in the nature of the organization, we cannot resolve this threat. However, we considered a variety of different product lines, and their characteristics in regard of the FEF should be comparable to other product lines. One exception for this may be the business dimension, as most product lines are intended only for internal use. While we have to be careful with generalizing our results, due to these threats, we did report all adaptations and limitations that we experienced. Also, these characteristics should not impact how to apply the FEF, which was our main goal in this study.

6.4 Conclusion Validity

We collaboratively designed our research methodology and employed action research to react to new insights by refining our methodology. Still, we could not analyze, synthesize or publish all of the available data, and had to abstract our insights. So, we have to be careful while interpreting our results, especially as they represent a single experience report. We aimed to share as many details as possible about the adaptation, insights, and artifacts (i.e., the questions in Appendix A) to allow other organizations and researchers to understand and replicate our work.

Similarly, due to confidentiality, we cannot provide all details, for instance, regarding the organization's tools, customers, data, guidelines, and processes. While this prevents an exact replication of our study, this is arguably a problem of any experience report in a real-world setting, and cannot be overcome. Still, we aimed to provide as many details on our method and the product lines' characteristics as possible. So, other researchers and organizations can adopt our method to their own needs, and we encourage them to share their experiences and compare them to ours to support or challenge our findings.

7 Related Work

We introduced cost models and scoping techniques in Section 2, as they are related to the FEF, and their concepts are important to understand this article. In this section, we focus on works that are related to our actual contributions. So, we discuss research that reports experiences on assessing the maturity of SPLE practices or on planning their adoption in practice.

7.1 Assessing SPLE Maturity

While we are not aware of a study that fully applied the FEF in an industrial context, Nazar and Rakotomahefa (2016) assess a small company based on the BAPO dimensions. So, their work is probably closest to ours. The authors combine observations, interviews, questionnaires, and document analysis to obtain data. Unfortunately, they do not report on their experiences of conducting the assessment, but focus on the actual outcome for the company. As such, our scope is different. We complement their work by providing insights into the details of how to apply an FEF assessment in a large organization and on multiple product lines.

Knauber et al. (2000) report their experiences of applying PuLSE (Bayer et al. 1999) in six small- and medium-sized companies. As PuLSE focuses not only on assessing the maturity or potential of a product line, the experiences reported cover various areas, for instance, technology transfer, modeling, and architecting. In contrast to us, Knauber et al. do not focus on the application of PuLSE itself as an assessment technique, providing few experiences on its benefits and problems. So, we considerably complement the insights on how to best employ a product-line assessment, including more detailed challenges and benefits that can be expected.

Ahmed and Capretz (2011) report their experiences of assessing the business dimension with an own maturity model. To this end, they investigate two companies and focus on the limitations and utilization of this model. Some of the insights relate to those we obtained in our study. For instance, we conducted interviews with stakeholders from different roles to obtain different insights, while Ahmed and Capretz state that they did not consider the roles of the participants, despite it being an important aspect. While related to our work, we considerable differ in various important aspects from them. We are concerned with another assessment method, report more detailed insights, and focus more on the challenges of applying an assessment, rather than the limitations of a newly introduced model.

Berger et al. (2020) investigate to what extent 12 organizations adopted SPLE practices and techniques. The authors focus on various aspects, including, for example, the concept of features, established traceability, the usage of variability modeling and a platform, and the extent of configurability. As such, this study can be seen as a high-level assessment of the maturity of SPLE in an organization. The authors highlight open issues that need to be addressed to foster adopting SPLE in practice. Arguably, this work is strongly connected to ours, as it provides a different perspective on the assessment of SPLE practices.

7.2 Planning SPLE Adoption

Koziolek et al. (2016) are concerned with adopting a cost model and domain analysis (e.g., feature modeling) for four product lines of an organization. Similar to our work, the authors report on the operationalization and tailoring of these techniques for the organization. Moreover, Koziolek et al. conducted interviews with stakeholders from different roles to collect their data. While few insights are similar to ours, for example, on establishing a common knowledge base on the products or product line under investigation, most of the insights are concerned with specifics of cost modeling and domain analysis—focusing on a risk assessment before adopting SPLE

practices. Consequently, our work is complementary by providing experiences on how to apply an FEF assessment on an existing product line.

Similarly, Rincón et al. (2019) assess to what extent their recently proposed APPLIES framework (Rincón et al. 2018) for deciding on the adoption of SPLE is useful in industry. Rincón et al. focus on similar research questions as we do, reporting to what extent their framework is perceived useful, what improvements the organization achieved, and what lessons they learned—based on two workshops with employees of the organization. Despite the similarities, major differences to our work are the focus on assessing the potential for product-line adoption and the consequent use of a different framework. In contrast, we were concerned with assessing the maturity of existing product lines using the established FEF.

Several other works report experiences of implementation, adoption, and scoping techniques for SPLE in practice (Ahmed et al. 2007; Bastos et al. 2017; Berger et al. 2014a; Böckle et al. 2002; Buhrdorf et al. 2003; Clements et al. 2001; Fogdal et al. 2016; Garcia et al. 2019; Ghanam et al. 2012; Hetrick et al. 2006; Jensen 2007; Krueger et al. 2008; van der Linden et al. 2007; Northrop 2002; da Silva et al. 2014). Unfortunately, none of these works reports how to best apply maturity evaluations. For example, van der Linden et al. and Fogdal et al. report case studies on the successes of SPLE, but the reasoning for adopting the corresponding practices and how to monitor or assess their status is not explained. Other researchers, such as Böckle et al. and Ahmed et al., focus on how to establish an SPLE culture and important success factors. These works can contribute to assessing different BAPO dimensions in the FEF, for instance, the business dimension by showing economic benefits or the organization dimension by highlighting necessary restructurings. We complement such works with our study, providing insights into how to assess the maturity of a product line.

8 Conclusion

We reported a multi-case study in which we used the FEF to assess the maturity of nine product lines at a large organization. As we are not aware of any experience reports of applying the FEF in practice, we believe these insights are highly valuable for organizations to assess their SPLE practices, and for researchers to analyze and improve on our findings. To this end, we explained why and how we tailored the FEF to the organization's domain and what challenges as well as benefits of applying the FEF we faced. Overall, we defined 67 questions, conducted 27 semi-structured interviews with various stakeholders of the product lines, wrote and revised reports, and performed workshops to collect our data. We received overwhelmingly positive feedback from the organization, which experienced immediate benefits that arguably exceed the expectations of simply assessing the maturity of a product line. As such, the FEF is a reasonable assessment to scope and plan the SPLE practices of an organization, despite the efforts needed to tailor and apply it to an organization.

Tailoring. The FEF (van der Linden et al. 2004, 2007) provides only a toy example to exemplify its usage. As no real-world studies exist, it is not surprising that we had to perform considerable adaptations to the rather abstract methodology. Most importantly, we found that building a common knowledge base, unifying terminologies, adapting questions, and involving different stakeholders is essential for a successful assessment. To elicit information and derive actions, we experienced

that synthesizing semi-structured interviews and reviewing the results with the stakeholders is valuable.

Challenges. Applying the FEF came with challenges. In particular, we experienced that it can be challenging to establish the knowledge required for the stakeholders to understand SPLE concepts that they did not distinguish, yet (e.g., domain versus application engineering). Moreover, it can be challenging to keep the focus on the actual FEF assessment if other problems prevail, and to align concepts of different software engineering practices. Not surprisingly, it can be problematic to justify the investments needed for a proposed action, and to assign such actions to the right stakeholder, if they do not align to the current organizational structure. Most interesting is the challenge of identifying problems that occur outside of the product line, but affect it. The particular example we experienced are task forces that work independently of the remaining architecture and may introduce unwanted changes. Identifying and addressing such problems is interesting future work.

Benefits. The benefits we experienced exceeded the expectations of a simple assessment. Applying the FEF automatically disseminated knowledge about SPLE practices and the product lines, which again connected stakeholders by providing a common body of knowledge. As intended, the FEF allowed us to identify shortcomings in the BAPO dimensions and to define road-maps as well as goals for each product line. Interestingly, we proposed actions that could be implemented immediately without a specific goal in mind, and still reduced the development and maintenance costs. This indicates that an FEF assessment is indeed a helpful means for an organization to improve its SPLE practice. However, managing the knowledge of an evolving software system and establishing a common ground for communication seems to remain a practical problem for future work.

Future Work. We plan to apply maturity assessments for software ecosystems (Berger et al. 2014b; Schultis et al. 2014; Seidl et al. 2017), since one of the assessed product lines had already taken first steps in that direction, and to compare the results to our FEF results to the extent possible. For some product lines, customers have indicated additional problems that we did not identify during the FEF assessment. So, we intend to expand the FEF with a customer perspective to also capture their experiences. **RQ$_2$** is an interesting question to investigate in more detail, for example, after five years of using the FEF for these product lines. We intend to do that and to find quantitative measures for the benefits of measuring the maturity of SPLE to complete the more qualitative measures presented in this article. Moreover, we plan to develop a decision model that managers can use as a dashboard with data from FEF measurements, supporting their decisions on the product lines and goals regarding the BAPO concerns. Finally, we are particularly interested in how to identify problems that affect, but are not part of a product line. For this purpose, we plan to conduct additional studies to understand more reasons where such problems stem from and how they could be avoided, or at least noticed.

A Appendix - Questions in the Interviews

Below is a selection of the questions we used during the interviews. We omit or changed questions that are specific to SAAB AB in this translation from Swedish. Questions marked with **M** were answered by managers, **T** by technical leads, and **E** by engineers.

    General Information
– What is your (the interviewee's) name? (M, T, E)

- What are your responsibilities and tasks? (M, T, E)
- What are the domain and your experiences from the domain? (M, T, E)
- In your own words, what is the purpose of the product line? (M, T, E)
- In your own words, who are the requirements stakeholders of the product line? (M, T, E)
- In your own words, what are the variants of the product line? (M, T, E)

Business - General

- Are the systems developed, planned, and marketed as single systems or as a product line? (M)
- Have you calculated the benefits of using SPLE? In time/effort, time-to-market, and investments? (M)
- Is there a budget for domain engineering? (M)
- Are the management and the organization aware of the costs and savings from using reusable assets? (M)

Business - Vision and Goals

- Is there a commitment from the management to work towards SPLE? (M)
- Would you say that there is support for SPLE in the organizational structure? (M)
- Would you say that your organization owns your vision for SPLE, or does it come from outside? (M)

Business - Planning

- Are the results of domain engineering taken into account when planning new products? (M)
- Are the plans for domain engineering and application engineering separate? (M)
- Is the planning for reusable components coordinated or controlled from a single product (application)? (M)
- Are the plans created so that they create the best business value overall? (M)

Architecture

- Describe the architecture of the products in the product line. Is there a platform that is common to all products in the product line? Which strategy is used? Plugins? Does the platform support late binding? Is it the same architecture in all products in the product line? (T, E)
- What strategies for inactive features are used? Are they included in the delivery, but disabled? How are they disabled? Is the feature removed? Are both, the feature and its interface, removed? (T, E)
- Is there a general platform that could be used in other product lines? (T, E)
- How is architectural integrity maintained to avoid architecture erosion?" (T, E)
- Does the architecture change in a controlled way to adapt to new requirements and circumstances? (T, E)
- Have you ever done a restart or redesign of the entire architecture? (T, E)
- Are you separating data and algorithms or otherwise trying to minimize the amount of duplicate code between variants? (T, E)
- How is the link between a variant (end product) and its input assets handled? Do you write on the asset to which final product it should belong? (M, T, E)
- Discussion on binding times: When are decisions about variability bound? At check out, build time (e.g., preprocessor), load time, run time (e.g., configuration files), ...? (T, E)
- How do you experience the relationship of a particular product variant to other product variants in the product line? As an adaptation, evolution, a variant? (M, T, E)
- Are there any assets that you use in the product line and that are also used in other product lines? (M, T, E)
- Where is the information about how components are interconnected? Is any type of standardized solution pattern (e.g., micro services) used? (T, E)
- Are the reusable components so mature in respect of reuse that you can build a whole new product of existing components without actually developing anything new (no new source code, etc.)? (M, T, E)
- Do you see that any part of the configuration process could be automated further? (M, T, E)

Process - General

- Describe the communication between domain- and application-engineering teams. Is it formal? For example, based on requirement documents, issues, etc.? Or is it informal, basically "go and talk to each other?" (M, T, E)

Process - Domain Engineering

- How do you handle conflicting requirements from two application engineering teams? (T, E)

– Do you manage assets that come from outside (other parts of the company, subcontractors, purchases, etc.)? (T, E)
– Are there processes for managing reusable components between multiple projects? (T)
– How do you handle the requirements for the products? Are there separate requirement documents for each product? Are the requirements labeled with the name of the final product/variant? (T)
– Do you feel that there is a requirements collection for the product line as a whole and not just for each product variant? (T, E, M)
– Is it possible to trace requirements between application and domain engineering? (T)
– Are all components developed with the ambition of being reusable in future products? (T, E)
– Do you measure how components are reused? Do you know where components are used? (M, T)
– Do you have metrics for measuring how well the development organization works? (M)
– Do you have improvement work based on measured values? Is this done continuously? (M, E)
– Are improvements in domain and application engineering done separately or together? (M, E)
– Can you briefly describe the process of improvement work? (M, E)

Process - Application Engineering

– When planning a new variant (end product), is there any person or role that has extra good knowledge about how components can be reused or linked together? Is that knowledge described in a document or in a formal language (with relationships, dependencies, etc.)? (T)
– Is it described in the process that new components developed for a product variant should be reusable for future use? (M, T, E)
– When discussing the requirements with customers (may be internal), do you talk about which reusable components already exist and which ones need to be developed? (M, T, E)
– Is there any training in how to work in shared reusable components and architectures? (M, T, E)

Process - Collaboration

– Are there any structures (organization, management, processes) that control how components can be reused? Or is it free for anyone who needs to use a component? (T, E)
– When new requirements apply to a product variant, do you use any process to review whether the requirements should be included in the product line and possibly affect all variants? (T, E)
– Is there a coordinated decision-making process on changes in shared components? (T, E)
– Is there a roadmap for future products, components, etc. that includes both possible future customer requirements, but also requirements to stay state-of-the-art in the area? (M)
– Are there separate processes for testing in domain and application engineering? Are they coordinated? (E)
– When planning and setting overall strategic plans for your area, do you take into account the future needs of individual product variants, or do you also consider shared components in domain engineering? (M)
– Are there responsibilities that span multiple product variants or are they limited to single variants? (M)
– Are risk identification and risk management performed on each product variant or is it also done on domain engineering? (M)
– Do you have a retirement/disposal plan for each shared component? (M, T)

Organization

– Are teams separated between domain and application engineering? (M)
– Do you have support from both project and line management for the separation? (M, T)
– Are there roles that are solely related to either domain or application engineering? (M)
– How easy/difficult is it to switch tasks between teams (specifically between domain and application engineering)? How often can you change? (M, T)
– Are you striving to create a common base, organizationally, on which many future products can be built? (M, T)
– Does the line organization support a product-line-adapted approach? (M)
– Does the technology organization support a product-line-adapted working method? (M)
– Where would you say that the greatest power lies in how products are developed? In the line management, in the projects or in the technical management? (M, T, E)

Additions

– Would you like to add anything? (M, T, E)
– What is your impression of the interview and of the FEF framework so far? (M, T, E)

References

Ahmed F, Capretz LF (2011) A business maturity model of software product line engineering. Information Systems Frontiers 13(4):543–560, DOI 10.1007/s10796-010-9230-8

Ahmed F, Capretz LF, Sheikh SA (2007) Institutionalization of software product line: An empirical investigation of key organizational factors. Journal of Systems and Software 80(6):836–849, DOI 10.1016/j.jss.2006.09.010

Ali MS, Babar MA, Schmid K (2009) A comparative survey of economic models for software product lines. In: Euromicro Conference on Software Engineering and Advanced Applications, IEEE, SEAA, pp 275–278, DOI 10.1109/SEAA.2009.89

America P, Obbink JH, van Ommering RC, van der Linden F (2000) CoPAM: A component-oriented platform architecting method family for product family engineering. In: International Conference on Software Product Lines, Kluwer, SPLC, pp 167–180

Andersson H (2012) Variability and customization of simulator products: A product line approach in model based systems engineering. PhD thesis, University of Linköping

Antkiewicz M, Ji W, Berger T, Czarnecki K, Schmorleiz T, Lämmel R, Stănciulescu Ş, Wąsowski A, Schaefer I (2014) Flexible product line engineering with a virtual platform. In: International Conference on Software Engineering, ACM, ICSE, pp 532–535, DOI 10.1145/2591062.2591126

Apel S, Batory D, Kästner C, Saake G (2013) Feature-Oriented Software Product Lines. Springer, DOI 10.1007/978-3-642-37521-7

Bastos JF, Neto PAdMS, O'Leary P, de Almeida ES, de Lemos Meira SR (2017) Software product lines adoption in small organizations. Journal of Systems and Software 131:112–128, DOI 10.1016/j.jss.2017.05.052

Bayer J, Flege O, Knauber P, Laqua R, Muthig D, Schmid K, Widen T, DeBaud JM (1999) PuLSE: A methodology to develop software product lines. In: Symposium on Software Reusability, ACM, SSR, pp 122–131, DOI 10.1145/303008.303063

Berger T, Guo J (2014) Towards system analysis with variability model metrics. In: International Workshop on Variability Modelling of Software-Intensive Systems, ACM, VaMoS, pp 23:1–8, DOI 10.1145/2556624.2556641

Berger T, Rublack R, Nair D, Atlee JM, Becker M, Czarnecki K, Wąsowski A (2013) A survey of variability modeling in industrial practice. In: International Workshop on Variability Modelling of Software-Intensive Systems, ACM, VaMoS, pp 7:1–7:8, DOI 10.1145/2430502.2430513

Berger T, Nair D, Rublack R, Atlee JM, Czarnecki K, Wąsowski A (2014a) Three cases of feature-based variability modeling in industry. In: International Conference on Model Driven Engineering Languages and Systems, MODELS, pp 302–319, DOI 10.1007/978-3-319-11653-2_19

Berger T, Pfeiffer R, Tartler R, Dienst S, Czarnecki K, Wąsowski A, She S (2014b) Variability mechanisms in software ecosystems. Information and Software Technology 56(11):1520–1535, DOI 10.1016/j.infsof.2014.05.005

Berger T, Lettner D, Rubin J, Grünbacher P, Silva A, Becker M, Chechik M, Czarnecki K (2015) What is a feature? A qualitative study of features in industrial software product lines. In: International Conference on Software Product Lines, ACM, SPLC, pp 16–25, DOI 10.1145/2791060.2791108

Berger T, Steghöfer J, Ziadi T, Robin J, Martinez J (2020) The state of adoption and the challenges of systematic variability management in industry. Empirical Software Engineering 25(3):1755–1797, DOI 10.1007/s10664-019-09787-6

Böckle G, Muñoz JB, Knauber P, Krueger CW, do Prado Leite JCS, van der Linden F, Northrop L, Stark M, Weiss DM (2002) Adopting and institutionalizing a product line culture. In: International Conference on Software Product Lines, Springer, SPLC, pp 49–59, DOI 10.1007/3-540-45652-X_4

Boehm BW (1984) Software engineering economics. IEEE Transactions on Software Engineering SE-10(1):4–21, DOI 10.1109/TSE.1984.5010193

Boehm BW, Brown AW, Madachy R, Yang Y (2004) A software product line life cycle cost estimation model. In: International Symposium on Empirical Software Engineering, IEEE, ISESE, pp 156–164, DOI 10.1109/ISESE.2004.6

Bosch J (2002) Maturity and evolution in software product lines: Approaches, artefacts and organization. In: International Conference on Software Product Lines, Springer, SPLC, vol 2379, pp 257–271, DOI 10.1007/3-540-45652-X_16

Buhrdorf R, Churchett D, Krueger CW (2003) Salion's experience with a reactive software product line approach. In: International Workshop on Software Product-Family Engineering, Springer, PFE, vol 3014, pp 317–322, DOI 10.1007/978-3-540-24667-1_24

Clements PC, Krueger CW (2002) Point / counterpoint: Being proactive pays off / eliminating the adoption barrier. IEEE Software 19(4):28–31, DOI 10.1109/MS.2002.1020283

Clements PC, Northrop LM (2002) Software product lines - Practices and patterns. Addison-Wesley

Clements PC, Cohen S, Donohoe P, Northrop L (2001) Control channel toolkit: A software product line case study. Tech. Rep. CMU/SEI-2001-TR-030, Carnegie Mellon University

Clements PC, McGregor JD, Cohen SG (2005) The structured intuitive model for product line economics (SIMPLE). Tech. Rep. CMU/SEI-2005-TR-003, Carnegie Mellon University

Czarnecki K, Grünbacher P, Rabiser R, Schmid K, Wąsowski A (2012) Cool features and tough decisions: A comparison of variability modeling approaches. In: International Workshop on Variability Modelling of Software-Intensive Systems, ACM, VaMoS, pp 173–182, DOI 10.1145/2110147.2110167

Dubinsky Y, Rubin J, Berger T, Duszynski S, Becker M, Czarnecki K (2013) An exploratory study of cloning in industrial software product lines. In: European Conference on Software Maintenance and Reengineering, IEEE, CSMR, pp 25–34, DOI 10.1109/CSMR.2013.13

Easterbrook S, Singer J, Storey MA, Damian D (2008) Selecting Empirical Methods for Software Engineering Research, Springer, pp 285–311. DOI 10.1007/978-1-84800-044-5_11

El-Sharkawy S, Yamagishi-Eichler N, Schmid K (2019) Metrics for analyzing variability and its implementation in software product lines: A systematic literature review. Information and Software Technology 106:1–30, DOI 10.1016/j.infsof.2018.08.015

Fogdal T, Scherrebeck H, Kuusela J, Becker M, Zhang B (2016) Ten years of product line engineering at Danfoss: Lessons learned and way ahead. In: International Systems and Software Product Line Conference, ACM, SPLC, pp 252–261, DOI 10.1145/2934466.2934491

García S, Strüber D, Brugali D, Di Fava A, Schillinger P, Pelliccione P, Berger T (2019) Variability modeling of service robots: Experiences and challenges. In: International Workshop on Variability Modelling of Software-Intensive Systems, ACM, VaMoS, pp 8:1–6, DOI 10.1145/3302333.3302350

Ghanam Y, Maurer F, Abrahamsson P (2012) Making the leap to a software platform strategy: Issues and challenges. Information and Software Technology 54(9):968–984, DOI 10.1016/j.infsof.2012.03.005

Hetrick WA, Krueger CW, Moore JG (2006) Incremental return on incremental investment: Engenio's transition to software product line practice. In: Symposium on Object-Oriented Programming Systems, Languages, and Applications, ACM, OOPSLA, pp 798–804, DOI 10.1145/1176617.1176726

Jensen P (2007) Experiences with product line development of multi-discipline analysis software at Overwatch Textron Systems. In: International Software Product Line Conference, IEEE, SPLC, pp 35–43, DOI 10.1109/SPLC.2007.18

Jørgensen M, Boehm B, Rifkin S (2009) Software development effort estimation: Formal models or expert judgment? IEEE Software 26(2):14–19, DOI 10.1109/MS.2009.47

Kalender ME, Tüzün E, Tekinerdogan B (2013) Decision support for adopting SPLE with Transit-PL. In: International Software Product Line Conference, ACM, SPLC, pp 150–153, DOI 10.1145/2499777.2499783

Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS (1990) Feature-oriented domain analysis (FODA) feasibility study. Tech. Rep. CMU/SEI-90-TR-21, Carnegie Mellon University

Khurum M, Gorschek T, Pettersson K (2008) Systematic review of solutions proposed for product line economics. In: International Software Product Line Conference, Lero, SPLC, pp 277–284

Klünder J, Hohl P, Schneider K (2018) Becoming agile while preserving software product lines: An agile transformation model for large companies. In: International Conference on Software and System Process, ACM, ICSSP, pp 1–10, DOI 10.1145/3202710.3203146

Knauber P, Muthig D, Schmid K, Widen T (2000) Applying product line concepts in small and medium-sized companies. IEEE Software 17(5):88–95, DOI 10.1109/52.877873

Koziolek H, Goldschmidt T, de Gooijer T, Domis D, Sehestedt S, Gamer T, Aleksy M (2016) Assessing software product line potential: An exploratory industrial case study. Empirical Software Engineering 21(2):411–448, DOI 10.1007/s10664-014-9358-0

Krueger CW, Churchett D, Buhrdorf R (2008) HomeAway's transition to software product line practice: Engineering and business results in 60 days. In: International Software Product Line Conference, IEEE, SPLC, pp 297–306, DOI 10.1109/SPLC.2008.36

Krüger J (2016) A cost estimation model for the extractive software-product-line approach. Master's thesis, Otto-von-Guericke University Magdeburg

Krüger J, Berger T (2020a) Activities and costs of re-engineering cloned variants into an integrated platform. In: International Working Conference on Variability Modelling of Software-Intensive Systems, ACM, VaMoS, pp 21:1–10, DOI 10.1145/3377024.3377044

Krüger J, Berger T (2020b) An empirical analysis of the costs of clone- and platform-oriented software reuse. In: Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ACM, ESEC/FSE

Krüger J, Fenske W, Meinicke J, Leich T, Saake G (2016) Extracting software product lines: A cost estimation perspective. In: International Systems and Software Product Line Conference, ACM, SPLC, pp 354–361, DOI 10.1145/2934466.2962731

Krüger J, Mahmood W, Berger T (2020) Promote-pl: A round-trip engineering process model for adopting and evolving product lines. In: International Conference on Systems and Software Product Line Engineering, ACM, SPLC

van der Linden FJ (2002) Software product families in Europe: The Esaps & Café projects. IEEE Software 19(4):41–49, DOI 10.1109/MS.2002.1020286

van der Linden FJ (2005) Family evaluation framework overview & introduction. Tech. Rep. PH-0503-01, Philips Medical Systems

van der Linden FJ, Bosch J, Kamsties E, Känsälä K, Obbink H (2004) Software product family evaluation. In: International Conference on Software Product Lines, Springer, SPLC, pp 110–129, DOI 10.1007/978-3-540-28630-1_7

van der Linden FJ, Schmid K, Rommes E (2007) Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Springer, DOI 10.1007/978-3-540-71437-8

Marchezan L, Rodrigues E, Bernardino M, Basso FP (2019) A customizable SPL scoping process for SPL reengineering. In: Anais da III Escola Regional de Engenharia de Software, SBC, pp 137–146

Montagud S, Abrahão S, Insfran E (2012) A systematic review of quality attributes and measures for software product lines. Software Quality Journal 20(3-4):425–486

Nazar N, Rakotomahefa T (2016) Analysis of a small company for software product line adoption–An industrial case study. International Journal of Computer Theory and Engineering 8(4):313–322, DOI 10.7763/IJCTE.2016.V8.1064

Nešić D, Krüger J, Stănciulescu Ş, Berger T (2019) Principles of feature modeling. In: Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ACM, ESEC/FSE, pp 62–73, DOI 10.1145/3338906.3338974

Niemelä E, Matinlassi M, Taulavuori A (2004) Practical evaluation of software product family architectures. In: International Conference on Software Product Lines, Springer, SPLC, pp 130–145, DOI 10.1007/978-3-540-28630-1_8

Northrop LM (2002) SEI's software product line tenets. IEEE software 19(4):32–40, DOI 10.1109/MS.2002.1020285

Olsson HH, Allahyari H, Bosch J (2012) Climbing the "Stairway to Heaven" - A mulitiple-case study exploring barriers in the transition from agile development towards continuous deployment of software. In: Euromicro Conference on Software Engineering and Advanced Applications, IEEE, SEAA, pp 392–399, DOI 10.1109/SEAA.2012.54

Pohl K, Böckle G, van der Linden FJ (2005) Software Product Line Engineering: Foundations, Principles and Techniques. Springer, DOI 10.1007/3-540-28901-1

Rincón L, Mazo R, Salinesi C (2018) APPLIES: A framework for evaluating organization's motivation and preparation for adopting product lines. In: International Conference on Research Challenges in Information Science, IEEE, RCIS, pp 1–12, DOI 10.1109/RCIS.2018.8406641

Rincón L, Mazo R, Salinesi C (2019) Analyzing the convenience of adopting a product line engineering approach: An industrial qualitative evaluation. In: International Systems and Software Product Line Conference, ACM, SPLC, pp 90–97, DOI 10.1145/3307630.3342418

Rosenmüller M, Siegmund N (2010) Automating the configuration of multi software product lines. In: International Workshop on Variability Modelling of Software-Intensive Systems, University Duisburg-Essen, VaMoS, pp 123–130

Schmid K, Verlage M (2002) The economic impact of product line adoption and evolution. IEEE Software 19(4):50–57, DOI 10.1007/3-540-28901-1

Schmid K, John I, Kolb R, Meier G (2005) Introducing the PuLSE approach to an embedded system population at Testo AG. In: International Conference on Software Engineering, ACM, ICSE, pp 544–552, DOI 10.1145/1062455.1062552

Schultis K, Elsner C, Lohmann D (2014) Architecture challenges for internal software ecosystems: a large-scale industry case study. In: International Symposium on Foundations of Software Engineering, ACM, FSE, pp 542–552, DOI 10.1145/2635868.2635876

Seidl C, Berger T, Elsner C, Schultis K (2017) Challenges and solutions for opening small and medium-scale industrial software platforms. In: International Systems and Software Product Line Conference, ACM, SPLC, pp 153–162, DOI 10.1145/3106195.3106203

Siegmund J, Siegmund N, Apel S (2015) Views on internal and external validity in empirical software engineering. In: International Conference on Software Engineering, IEEE, ICSE, pp 9–19, DOI 10.1109/ICSE.2015.24

da Silva IF, da Mota Silveira Neto PA, O'Leary P, De Almeida ES, de Lemos Meira SR (2011) Agile software product lines: A systematic mapping study. Software: Practice and Experience 41(8):899–920, DOI 10.1002/spe.1078

da Silva IF, da Mota Silveira Neto PA, O'Leary P, De Almeida ES, de Lemos Meira SR (2014) Software product line scoping and requirements engineering in a small and medium-sized enterprise: An industrial case study. Journal of Systems and Software 88:189–206, DOI 10.1016/j.jss.2013.10.040

Stănciulescu Ş, Schulze S, Wąsowski A (2015) Forked and integrated variants in an open-source firmware project. In: 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME), IEEE, pp 151–160

Thummalapenta S, Cerulo L, Aversano L, Di Penta M (2010) An empirical study on the maintenance of source code clones. Empirical Software Engineering 15(1):1–34, DOI 10.1007/s10664-009-9108-x

Thurimella AK, Padmaja TM (2014) Economic models and value-based approaches for product line architectures. In: Economics-Driven Software Architecture, Elsevier, pp 11–36, DOI 10.1016/b978-0-12-410464-8.00002-7

Tüzün E, Tekinerdogan B, Kalender ME, Bilgen S (2015) Empirical evaluation of a decision support model for adopting software product line engineering. Information and Software Technology 60:77–101, DOI 10.1016/j.infsof.2014.12.007

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer, DOI 10.1007/978-3-642-29044-2

Yin RK (2003) Case Study Research: Design and Methods. Sage

Robert Lindohf is a principal engineer at Saab Aeronautics Simulation Center. His interests include product line engineering, change leadership, software development, technical training etc. In his professional role at Saab he works at the Technology Management Office with coordination of technology management and development strategies for simulator development. Robert has presented the PLE approach of Saab Aeronatics Simulation Center at the Nordic System Engineering Tour and at the Royal Aeronautical Society.

Jacob Krüger is a PhD student and associated researcher at the Databases and Software Engineering work group of the Otto-von-Guericke University Magdeburg, Germany. He received his M.Sc. degree in Business Informatics at the Otto-von-Guericke University in 2016, has been working as research associate at the Harz University of Applied Sciences Wernigerode, Germany, and visited Chalmers University of Technology | University of Gothenburg, Sweden, as well as the University of Toronto, Canada. His research focuses on feature-oriented software development, with particular interests on software evolution, program comprehension, and human factors.

Erik Herzog is a Technical Fellow at Saab Aeronautics and a Certified Systems Engineering Professional (CSEP). He received his Ph.D. at the Department of Computer and Information Sciences at Linköping University, Sweden. His interests include development and introduction of Systems Engineering processes, with special emphasis on requirements engineering, architecting and integration, information modelling, tool integration techniques and change leadership. In his professional role at Saab he adapting overall systems development methodology for major development programmes such as Gripen E.
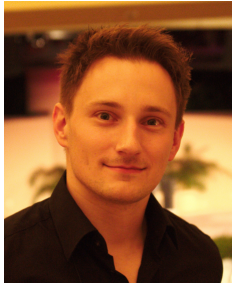
Thorsten Berger is an Associate Professor at the department of Computer Science and Engineering at the University of Gothenburg and Chalmers University of Technology. His research focuses on model-driven software engineering, program analysis, and empirical software engineering—to build methods and tools for increasing the automation in software engineering. He received his Ph.D. from the University of Leipzig in 2013, then worked as a Postdoctoral Fellow at the University of Waterloo in Canada and the IT University of Copenhagen in Denmark. He published in major software-engineering conferences (e.g., ICSE, FSE, ASE, OOPSLA) and journals (e.g., IEEE TSE, Springer EMSE, IEEE Software), and received best-paper awards at the 2015 ACM SIGPLAN conference on MODULARITY and the 2013 European Conference on Software Maintenance and Reengineering (CSMR, now IEEE SANER). He serves in the program committees of major conferences, including ICSE, FSE, and ASE, recognized with distinguished reviewer awards at ICSE 2020 and ASE 2018. He received grants from competitive funding agencies (e.g., Swedish Research Council) and was awarded one of the most competitive Swedish grants, the Wallenberg Academy Fellowship.