Not Quite There Yet: Remaining Challenges in Systems and Software Product Line Engineering as Perceived by Industry Practitioners

Martin Becker
Embedded Systems
Fraunhofer IESE
Kaiserslautern, Germany
Martin.Becker@iese.fraunhofer.de

Rick Rabiser
CDL VaSiCS, LIT CPS Lab
Johannes Kepler University Linz
Linz, Austria
rick.rabiser@jku.at

Goetz Botterweck
Trinity College Dublin and Lero - the Irish Software Research Centre
Dublin, Ireland
goetz.botterweck@tcd.ie

ABSTRACT

Research on system and software product line engineering (SPLE) and the community around it have been inspired by industrial applications. However, despite decades of research, industry is still struggling with adopting product line approaches and more generally with managing system variability. We argue that it is essential to better understand why this is the case. Particularly, we need to understand the current challenges industry is facing wrt. adopting SPLE practices, how far existing research helps industry practitioners to cope with their challenges, and where additional research would be required. We conducted a hybrid workshop at the 2023 Systems and Software Product Line Conference (SPLC) with over 30 participants from industry and academia. 9 companies from diverse domains and in different phases of SPLE adoption presented their context and perceived challenges. We grouped, discussed, and rated the relevance of the articulated challenges. We then formed clusters of relevant research topics to discuss existing literature as well as research opportunities. In this paper, we report the industry cases, the identified challenges and clusters of research topics, provide pointers to existing work, and discuss research opportunities. With this, we want to enable industry practitioners to become aware of typical challenges and find their way into the existing body of knowledge and to relevant fields of research.

CCS CONCEPTS

• **Software and its engineering** → **Software product lines**.

KEYWORDS

Software product line engineering, industry challenges

ACM Reference Format:
Martin Becker, Rick Rabiser, and Goetz Botterweck. 2024. Not Quite There Yet: Remaining Challenges in Systems and Software Product Line Engineering as Perceived by Industry Practitioners. In *28th ACM International Systems and Software Product Line Conference (SPLC '24), September 02–06, 2024, Dommeldange, Luxembourg*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3646548.3672587

SPLC '24, September 02–06, 2024, Dommeldange, Luxembourg

ACM ISBN 979-8-4007-0593-9/24/09
https://doi.org/10.1145/3646548.3672587

1 INTRODUCTION

System and Software Product Line Engineering (SPLE) has a long tradition, tracing back almost half a century to Parnas' discussion of program families [59]. From the get-go frameworks for SPLE [17], the system and software product line community, and its flagship conference, SPLC[1] have been very much inspired by industry needs and challenges. The devised approaches have been applied successfully in a broad spectrum of industries [51, 80].

However, despite all this research, when talking to industry practitioners, one gets the impression that they are still struggling when dealing with SPLE in practice. One common pattern is that a company with a successful product adds more similar products in an ad hoc approach (e.g., "clone-and-own"). Then, with more products, at some point the company hits a complexity barrier and requires a more systematic approach [10, 24, 69]. On the other hand, the combinatorial explosion resulting from developing highly-variable engineering artefacts leads to issues in other engineering practices, e.g., verification and validation/testing, if no systematic approach to manage variabilities is applied [55] and variability management across granularity levels is perceived as being difficult [26]. The evolution of variation-rich system without a systematic approach also becomes a nightmare [41] over time.

In this paper, we argue that, to provide directions for product line research, it is necessary to understand this phenomenon and reasons for it. In 2020, Berger et al. provided some updates on industry challenges in SPLE [10] elicited earlier, but overall there is only limited up-to-date data on industrial challenges in SPLE. Single case studies about companies adopting product line engineering are still added to the body of knowledge, see, for instance, the ESPLA catalog [51, 52], but multi-case studies or studies focusing on discussing industry challenges from a broader perspective often date 10 to almost 20 years back [9, 80].

Furthermore, SPLC industry participation keeps eroding, e.g., Schmid et al. [73] concluded in 2021 that "today, SPLC is mostly regarded as an academic conference with little industry participation". However, they also emphasised the interest of practitioners and researchers to exchange knowledge and learn from each other.

We argue that industry practitioners should (1) be aware of typical challenges in adopting SPLE practices and (2) understand the existing body of knowledge on SPLC practices and technology. Furthermore, the SPLC research community needs to (3) understand *current and unsolved* challenges that industry practitioners are (still) facing and (4) try to address the challenges in collaboration with

[1] https://splc.net/

industry practitioners. This, we hope, will also re-strengthen the link between the academic and industry subcommunities.

As a first step, two authors of this submission organised a workshop with industry practitioners from 9 companies with active systems and software product lines. The discussion and data emerging from that workshop provide the input for this paper. With our analysis here, we aim to answer the following research questions:

RQ1 (Experienced Challenges). Which challenges and barriers do industry practitioners experience in system and software product line practice?

RQ2 (Reasons). What are the reasons for these observed challenges and barriers? (a) Where do approaches exist but are not applied (indicating dissemination, communication, or management issues); (b) where are the available approaches insufficient (indicating opportunities for future work)?

RQ3 (Way Forward). Where are opportunities and what are appropriate measures to (re-)strengthen the synergy between academic SPLE researchers and industrial practitioners?

In summary, this paper provides the following **contributions**:

- Recent and up-to-date empirical data elicited from practitioners working on 9 different industrial use cases, including 17 challenges organised into 7 themes.
- A rating of the relevancy (i.e., the relative importance as rated by practitioners) of these challenges.
- A discussion of existing research to address the challenges and open research opportunities for the SPLE community.

2 RESEARCH APPROACH

In this section, we summarise our research approach to obtain the presented results (see Figure 1). The workshop was communicated explicitly as an industry-focused and hybrid event (to reach a broad international audience, including those who might not travel to Tokyo). 33 participants took part – the vast majority joined online.

Workshop. ❶ Initially, industry participants presented their 9 cases (Section 3). We used a Miro board [54] to share material and intermediate results, which we cannot publish due to confidentiality. Instead, we extract the most relevant information here. We collected challenges, best practices, research ideas, and helpful publications for each case. From these 9 per-case collections we extracted and merged all challenges, resulting in 17 unique challenges.

Discussion in Workshop and Online. ❷ To produce a shared understanding and identify the most relevant challenges, we started a discussion and rating process in the workshop, which continued asynchronously after the workshop. Here, participants evaluated challenges by assigning 0 to 5 points (low to high relevance). The resulting challenges were then clustered into 7 themes by the organisers of the workshop (details in Section 4).

Review by Researchers. ❸ The next step was a review of the identified themes among researchers. For the further analysis we took inspiration from Thematic Analysis and related approaches [14, 79], in particular from guidelines for the application of Thematic Synthesis in Software Engineering research suggested by Cruzes and Dybå [18]. In summary, they suggest to follow five phases:

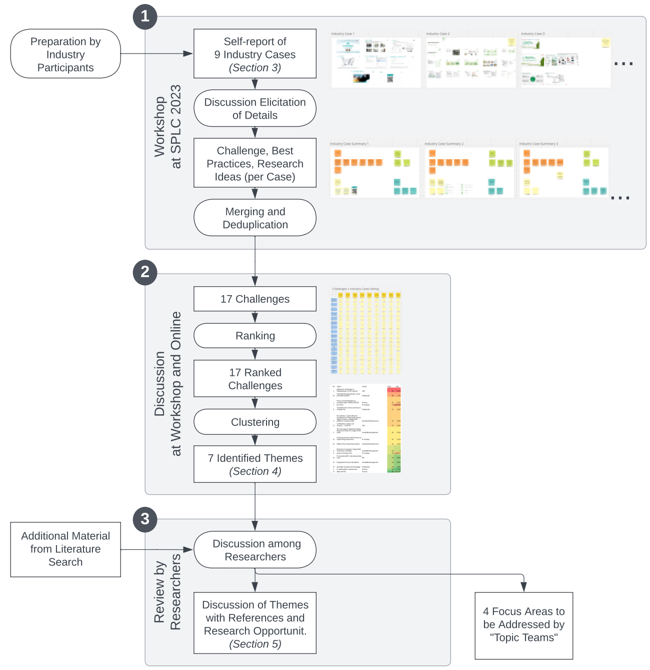

Figure 1: Overview of the Research Approach.

- Extract data – Achieve immersion to be familiar with the evidence; update the research protocol; extract data.
- Code data – Identify and code interesting concepts, categories, and findings across the entire dataset.
- Translate codes into themes – Translate codes into themes, identify subthemes and higher-order themes where necessary.
- Create a model of higher-order themes – Explore relationships between themes and create a model of higher-order themes.
- Assess the trustworthiness of the synthesis – Assess the trustworthiness of the interpretations leading up to the synthesis.

Note that thematic analysis/synthesis originally was invented for the analysis of a body of literature. Still, some of the provided ideas were useful to guide our research. As an output, we produced a mindmap with the identified themes and discuss research opportunities. More details later in Sections 4 and 5.

3 INDUSTRY CASES

In this section, we introduce the 9 industry cases presented at the workshop. For a condensed summary see Table 1.

The cases reported a diverse spectrum of *types* of product lines (Software PL, Hardware PL, System PL, System-of-Systems PL). 8 cases identified their project as "ongoing", one as "future".

3.1 Industry Case 1

Company. The first industry case, an automotive mechatronics expert, is adopting SPLE at both the software and system levels for several years now to cater to a diverse range of customers and a broad product portfolio.

Approaches. The company utilises a "docs as code" methodology, enabling streamlined updates and maintenance of documentation

Table 1: Overview of the nine industry cases.

Case	Domain	Focus	Main Challenges
1	Automotive mechatronics	Broaden product portfolio	Standards, "docs as code", CI/CD, test strategy, project-based org.
2	Sensors, measuring devices	Variant management & product configuration	PL verification
3	HVAC, home applicances	System of systems PLE	Ecosystem, multiple domains, verification and validation effort for variants, portfolio vs. engineering across distributed locations
4	Metallurgical plant solutions	Systematic variability management	Dependence on automation platform vendors, variability on multiple levels and in multiple disciplines
5	Agritech	Multi systems PL	Long-living systems, small production volumes, multiple product lines, maintenance, tools for modeling and simulation
6	Rail transport, rolling stock	Enhance reuse rate	Documenting, scoping and evaluating PL architectures, var. modeling, standardising modules, module maintenance, org. structures
7	Industrial automation	New generation of automation products	Knowledge silos, integration and testing, HIL testing, perception of slow platform development, modification of shared assets
8	Defense, aerospace	Increase modularisation and reuse	Perception of PLE, design authority, asynchronous information, governance, reuse scope, proactive reuse identification, frequent analysis
9	Automotive powertrain controllers	PL variant management	Consistent variant management across disciplines, variant management in V & V, evolution of PLs, PL of PLs, efficient var. realisation, usability of var. management tools, collaboration with OEMs

akin to software code. The configuration of features within the PL is managed through KConfig, that raises questions regarding its efficacy and usability for complex feature configurations in a system PL environment, including difficulties in maintaining an overview of features and configuring them.

Challenges. They have to adhere to the Automotive SPICE standard, necessitating robust management of variability across all work products. Furthermore, in the realm of Continuous Integration and Delivery (CI/CD), they are compelled to provide quick feedback cycles, implement rigorous quality gates, and satisfy the varying acceptance criteria of developers, customers, and projects. A crucial challenge is formulating a viable test strategy compatible with PL characteristics and efficiently addressing stakeholder needs. Additionally, there is a pressing need to restructure the organisation to facilitate a smooth transition from traditional project-based workflows to a more scalable product line engineering framework.

3.2 Industry Case 2

Company. The second case is an international company with subsidiaries in 7 countries plus 60 countries with distributors. They develop and manufacture sensing elements, sensing modules and sensors for air velocity, CO_2, humidity, moisture in oil, pressure, temperature, etc. They are maintaining a product line of measurement technology solutions such as (hand-held) measuring devices, humidity calibrators or wireless sensor systems. Their products are highly customisable and there is a massive amount of possible combinations of the components of the product line.

Approaches. The company has developed their own variant management and configuration tool, including description of features and their characteristics, as well as constraints on possible selections of features and characteristics.

Challenges. One of the core challenges they reported was the difficulty in performing verification, which is important in their domain, given the combinatorial explosion of possible products and also to automatically check the validity of created configurations. They have tried approaches such as equivalence class partitioning, but found no vectors since every single characteristic behaves (can behave) different in combination with others.

3.3 Industry Case 3

Company. The third case, an established OEM and supplier in various application domains, is advancing its mission to improve smart living solutions at home by implementing a System of Systems (SoS) PLE approach based upon its existing PLs, including electric solutions, climate & well-being etc. Their SPLE strategy focuses on managing resources effectively, supporting the definition of core products and assets that are foundational to multiple applications. The modularisation of products into manageable units is a key strategy, enabling them to tackle complex systems.

Approaches. Clarifying what constitutes a "module" in the context of SoS and establishing clear guidelines for modular development and integration are essential steps. This ensures that modular components are effectively designed and that they contribute to the overall objectives of the SoS architecture, facilitating smoother transitions and more robust system integrations.

Challenges. They are facing several challenges in this transformation described along the BAPO framework [80]. Business: transitioning from offering single interconnected systems to an integrated ecosystem. Architecture: managing a diverse portfolio and engineering processes distributed across locations while concurrently handling multiple domains (software, electronics, mechanics). Process: reducing the V&V effort required for variants across different product lines. Organisation: coordinating portfolio and engineering management across distributed locations. To evolve from System to SoS PLE, they consider strategic personnel transfers between different application domains to foster cross-functional expertise and collaboration.

3.4 Industry Case 4

Company. Case 4 is an international company providing metallurgical plant solutions. They maintain several SPLs (automation, control) for machines in the metallurgical process, e.g., ironmaking, steelmaking, continuous casting, hot rolling, cold rolling, and strip processing. Their motivation for systematic variability management stems from the need to consider different kinds of plants and customer requirements for those plants; the fact that every plant has several setups; that there are diverse protocols and diverse suppliers, e.g., of sensors; that there are many interfaces between systems (their own and of other vendors) and software on different levels of the automation pyramid. The company has developed diverse solutions to deal with variability including templates (typicals), standard technological packages, and parameterisation tools.

Approaches. The company has a long-term research project with multiple university partners to work on these challenges and adopt an SPL approach. Specifically, they are working on a multi-disciplinary variability modeling approach allowing to represent variability knowledge from different engineering disciplines in different variability models and relate them via cross-discipline constraints.

Challenges. The core challenges the company reported are the issues resulting from clone-and-own (e.g., code duplication, maintenance issues), the huge system scope, the need to adapt software to diverse hardware platforms (and the need to maintain legacy software for many years), the versioning of variants (customer configuration vs. platform), and the required (re-)modularisation to ease systematic variability management.

3.5 Industry Case 5

Company. The fifth case is an OEM of farming systems who manages diverse product families and maintains long-living systems, often exceeding 30 years in operation. Cybersecurity is a critical concern as it necessitates ongoing software updates to address emerging threats over the lifespan of the hardware.

Approaches. Addressing this concern involves integrating advanced tools for modelling and simulation, fostering a modular architecture that allows for component reuse, and continuously updating cybersecurity measures to protect against evolving threats. The reuse of components across different PLs is a strategic approach to maximise resource efficiency and reduce development costs.

Challenges. Small volumes complicate standardisation and economies of scale. A significant challenge arises when hardware required for testing new software versions is no longer available. This scenario demands innovative solutions such as virtualisation or emulation technologies, ensuring new software can still operate on legacy systems. Tool support for modeling and analysing variability across both space (hardware and system configurations) and time (across system versions and over time) is crucial. These tools have to manage complexity and ensure consistency and reliability across PLs. Moreover, the maintenance of long-living products requires robust strategies to ensure their functional safety and security against cyber threats throughout their extended lifespan.

3.6 Industry Case 6

Company. Our sixth case, an OEM for railway systems, is aiming to enhance its reuse rate through the adoption of modular architectures and building kits tailored for different market segments. The company operates in a market characterised by a relatively small number of customers, where evaluation criteria can vary significantly, even with the same customer. This variability affects bid types, which range from system-only proposals to comprehensive turnkey solutions.

Approaches. Their strategic shift is designed to improve economies of scope, accelerate the bid process, and expedite project execution while providing value-adding flexibility. The modular approach integrates components from various engineering disciplines (electrical, mechanical, networking, software).

Challenges. Even with the broad experience with reuse on all levels of the product structure there are several challenges and strategic questions: how can the company characterise demand variability and adapt product definitions to accommodate such changes? What criteria should be used to delineate a product family, especially when focusing on module-based families? What are the optimal criteria for defining modules and the next level of standardisation related to these modules? How can modularity be assessed efficiently to ensure it meets the desired outcomes and contributes to overall business objectives? How to effectively document a product family architecture that integrates functional, electrical, mechanical, network, and software components while balancing market/customer needs with technical and business objectives? How to develop subsystem variability models that effectively connect with train-level variability models? What strategies should be employed to optimise module development and maintenance? What is the most effective organisational and governance setup to manage modularity?

3.7 Industry Case 7

Company. Our seventh case is a tier-1 supplier in industrial automation, specialising in control systems for electrical motors in high-performance applications. They are transitioning to a new generation of automation products for a diverse range of motor sizes. With a strong foundation in PLE, the company seeks to overcome the sluggishness perceived in the previous product generation's PLE process, which was criticised for being "dead by roadmap."

Approaches. To speed up delivery of new features, the company has integrated agile methodologies within a CI/CD framework. Each product team manages its own backlog, which is overseen by a Product Owner. Coordination among these backlogs, especially for common features requiring platform team collaboration, is managed through a product owner forum. Additionally, a Governance Board is responsible for setting targets for software platform development, prioritising mid-to-long-term and complex platform cross-cutting development items, and ensuring that resources are adequately allocated. Product teams are permitted to modify shared assets, raising questions about the long-term impact of these changes on system stability.

Challenges. The company maintains solid internal and external interfaces to enhance platform stability, but faces cost pressures that could compromise critical components, threatening the controlled flexibility essential for its PL success. The balance between rapid development and high-quality, stable releases continues to be a pivotal concern in the competitive field of industrial automation. Also, product teams have limited awareness of other products and their software/hardware configurations. There are ongoing concerns about maintaining system consistency during CI/CD, including decisions on test selection (for each pull request) and test duration. Decisions on which hardware configurations and power sizes to test and appropriate configurations for each pull request remain complex. Platform development is perceived as slow due to long lead times for some changes due to backlog prioritisation.

3.8 Industry Case 8

Company. Our eighth case is a tier-1 defense supplier specialising in networked, pre-integrated sensor solutions with applications across ground, air, sea, cyber, and space platforms. The company, which operates on a global scale, manages several product families and variants, maintaining core assets across these groups. Driven by the need for efficient resource utilisation, timely delivery, and cost reductions, the company initiated a strategy focused on increasing modularisation and reuse across its divisions and business units. The primary motivation for this was the management of numerous contracted products under limited resources. The goal was to facilitate modularisation and implement a reuse potential analysis (RPA) to standardise solutions across different product families.

Approaches. Effective PLE implementation requires robust governance from top-level management. For instance, management needs to clearly define the reuse scope, which should extend from product families across business units and international entities. Reuse potentials are identified in a proactive manner, as starting the search for reuse opportunities strategically before developing new products can save significant effort. Regular reviews of product evolution are conducted to spot early reuse opportunities, especially concerning new requirements, technologies, and obsolescence. The RPAs are conducted collaboratively across business units and engineering departments, with regular updates and information sharing through a unified platform. This approach helps in making informed, timely decisions, thus avoiding protracted analyses. Additionally, appointing technical counterparts to product architects is essential for evaluating solutions that are applicable across product families. Finally, the consolidation and effective communication of information regarding target products, variants, features, and solution architectures are critical for maintaining alignment across the organisation.

Challenges. They are facing several challenges: as RPAs initially just were seen as an engineering effort, it required expanded governance to include management and business units to foster strategic reuse. Establishing a design authority is crucial for balancing product evolution with the objectives of modularisation and reuse. RPA revealed that information was often out of sync and varied in granularity, leading to the need for a standardised collaboration platform to ensure consistency in understanding.

3.9 Industry Case 9

Company. Our last case is a supplier of electronic controls for automotive powertrain solutions, with extensive experience with PLE. They manage over 3000 software deliveries annually, utilising a software platform governed by a central variability model.

Approaches. The model incorporates five distinct parts: market variants, context variants (including components and topologies), feature variants, technology variants (pertaining to hardware), and implementation variants (related to software). The process to derive specific project functions (100%) from the platform involves extending the platform functions (150%) with project-specific elements, followed by automatic removal of unselected parts. This is facilitated by a seamless management of system, hardware, and software variants, which includes more than 20,000 software variation points and more than 800,000 calibration parameters.

Challenges. Several challenges arise in managing such a complex hierarchical PL: achieving consistent and efficient variant management across multiple engineering disciplines is crucial. This includes maintaining artefact consistency across the variant management process. Managing variants effectively during V&V and continuous development stages presents logistical and technical difficulties. The evolution of variant models, especially concerning updates and cloud functionality integration, requires ongoing adaptation and flexibility. Managing "PLs of PLs" introduces complexity, requiring scalable and robust variant management solutions. Efficient handling of variants at the pre- and post-compilation stages is necessary to optimise performance and resource allocation. Enhancing the usability of variant management tooling, including tailoring for specific scopes, is essential for effective operation. Collaboration with OEMs, particularly concerning standardisation efforts, is vital for the harmonisation of practices and enhancing mutual benefits.

4 IDENTIFIED CHALLENGES AND THEMES

We first discuss challenges and their relevancy as perceived by practitioners and then identify theme clusters to structure our further discussion. Note that this is based both on material presented at the workshop *and* discussion after the workshop among participants and researchers to interpret and evolve the information.

4.1 Challenges

During the workshop we identified 17 challenges. We discuss them below, ordered by rating by the industry participants, highest average relevancy rating first (see Figure 2).

Efficient Product Line Verification & Validation, (especially in a CI/CD context) received the highest average rating and was a recurring topic at the workshop. While all the companies develop and maintain highly variable systems successfully, they struggle when

No	Topics	Cluster	Total	Average
4	Efficient PL Verification & Validation (esp. CI/CD context)	V&V	35	4.38
10	Modularization/granularity, (reuse potential) analysis	Modularity	30	3.75
1	Process standardization(e.g. company-wide, ASPICE) and PLE	Process	27	3.38
7	PLs of PLs	PL Scoping	27	3.86
8	Transition from Clone and Own to strategic PLE	Modularity	27	3.38
9	PL Evolution / system lifecycle management of long living systems (legacy systems, changing HW platforms, CyberSecurity)	Evolution&Maintenance	27	3.38
11	Certifications (safety and security, ...) and PLE	V&V	27	3.38
3	VM tool support (integrated,holistic (ReqS-Arch-Code-VV), usage of OSS tools)	VariabilityManagement	25	3.13
15	Involving business into PLE (from an engineering perspective)	PL Scoping	25	3.13
16	MBSE & PLE & long living systems	Evolution&Maintenance	25	3.13
2	Diversity of customers /Supporting unforeseen variants	VariabilityManagement	24	3.00
6	System of Systems PLs	PL Scoping	24	3.43
12	PL Evaluation(KPI, maturity,business case)	KPI	23	2.88
13	Integrated VM across disciplines	VariabilityManagement	23	2.88
14	SharingPL architectural knowledge	Modularity	23	2.88
5	PL organizations / governance	Process	22	2.75
17	Agile and PLE	Process	20	2.86

Figure 2: Topics identified in the workshop and their average relevancy rating defined by the industry case companies.

it comes to dealing with validating and particularly verifying the plethora of possible variants. Often, some standard configurations are tested and then a particular configuration for a customer, which is often inefficient manual work. Several companies reported their own custom-developed solutions to help them automate this. Yet, testing all possible configurations (or at least the most relevant ones) remains a big challenge.

Modularisation (granularity, reuse potential analysis) was also discussed by all companies as very important yet challenging. Specifically, companies agree that to properly adopt SPLE, one needs to properly modularise the (architecture of) systems. Due to the need to deal with long-living legacy systems and a lack of resources, refactoring everything systematically is just not feasible. Also, even if enough resources would be available, it remains challenging to hit the right level of granularity of modularisation (e.g., from feature-level or component-level down to #ifdef and statement level). For several companies, support to analyse the reuse potential of (parts of) their systems would be useful.

Process standardisation (e.g., company-wide, ASPICE [82]) *and PLE* was mentioned as both important and challenging. Customers and regulations require process standardisation. With adopting SPLE processes, this also requires standardisation of processes such as domain engineering and creating reusable components.

Product Lines of Product Lines was mentioned by companies as both a fact (it is what they do) and as a challenge. Most approaches companies have found focus on a single product line, and that mostly only in one particular discipline.

Transition from clone-and-own to strategic PLE is what has attracted companies to PLE in the first place. Companies not only naturally start with clone-and-own approaches to reuse, even after adopting a product line approach, they still (have to) follow some clone-and-own approaches, e.g., deriving a first version of a product from the product line for a customer and then cloning and adapting it for several similar customers. Both is challenging: adopting PLE in the first place and, after having adopted it, still living with customers having very specific requirements not allowing for a very high degree of standardisation. High enough to make PLE useful, but not high enough to get rid of clone-and-own completely.

PL Evolution / system lifecycle management of long-living systems (legacy systems, changing hardware platforms). Companies have very long living systems, e.g., company 4 reported their deployments live up to 30 years. Also, companies have to integrate their systems with systems developed by others, even competitors, frequently. Hardware platforms are provided by different vendors and change frequently resulting in significant adaptation efforts. Changing customer requirements, together with the very long lifecycle and changing hardware platforms make PL evolution a big challenge in practice.

Certifications (e.g., safety and security) and PLE are just a necessity for several of our case companies. Certification authorities, however, do not make it particularly easy to certify reusable components that cannot directly be run and tested until a product has been derived. This makes certification challenging in PLE scenarios.

Variability management tool support (integrated, holistic (requirements, architecture, code, V&V), usage of open-source tools). Companies require tools that help them manage variability throughout all development phases, from requirements over architecture, to code, to V&V, and back, integrated with their (application) lifecycle management tools. They are open to adopt open-source tools, yet often made the experience that the lifecycle support is even more limited than in commercial tools. Overall, they find it challenging to use existing variant management tools, because it is difficult to integrate them into their processes. Some of the companies, however, are actively using variant management tools such as the commercial pure::variants [63] and the open-source FeatureIDE [28]. Yet, "full" process integration is a different, challenging story. Hence, several of the companies have also developed their own solutions.

Involving business into PLE (from an engineering perspective). PLE is often perceived as an engineering effort. However, without involving business, it is impossible to maximise the benefit of systematic reuse. Companies often struggle with that because adopting a PLE approach is motivated and triggered by technical challenges in engineering such as maintenance and evolution effort when following clone-and-own approaches.

MBSE & PLE & long-living systems. Adopting model-driven software engineering approaches, as some of our case companies are doing, leads to the challenge that now models and meta-models become part of long-living systems in a PLE context. Dealing with these additional artefacts, and especially also their consistency, thus becomes challenging.

Diversity of customers / supporting unforeseen variability. Several of our case companies have a very diverse customer base, which in turn often leads to unforeseen (as in cannot be proactively planned) variability in their systems.

System of Systems PLs. The companies deal with systems of systems, i.e., their product line consists of multiple systems on different layers, e.g., of the automation pyramid, developed by different departments/companies, from different disciplines, without a strong SE background. This makes systematic approaches to variability management, across systems and disciplines, even more important.

PL Evaluation (KPIs, maturity, business case). All companies saw it as essential to not "just" adopt a PL approach, but also be able to assess it, e.g., from viewpoints of efficiency, performance, costs, etc. Measuring PL KPIs and, more generally, assessing the maturity of company when it comes to SPLE and its business cases, is both interesting and challenging in practice.

Integrated VM across disciplines. Integrating the different perspectives on variability the different disciplines have is challenging, especially when considering the different backgrounds, notations, and tools used by mechanics, electronics and electrics, software engineering, hydraulics, and many more.

Sharing PL architectural knowledge remains challenging in practice. Companies report that such knowledge about their highly variable systems is often concentrated in few people and that no single person has a complete overview.

PL organisations / governance. Companies struggle with the urge to centralise domain and platform engineering (development for reuse) vs. engineering departments being internationally distributed and involved in multiple projects.

Agile and PLE. All of our case companies to some extent have adopted agile development approaches. Naturally, they also aim for agile approaches to product line engineering. Looking at most, especially early works on SPLE, however, makes it appear not "lightweight" enough to stay agile at the same time.

Figure 3: Identified Themes.

Answering RQ1 (Experienced Challenges)
Reported challenges are (in descending importance): efficient PL verification and validation; modularisation, process standardisation; multi PLs and SoS PLs; transitioning from clone-and-own to PLE; PL evolution and system lifecycle management; certifications; tool support; involving business into PLE; MBSE and PLE; the diversity of customers and emerging variability; PL evolution and KPIs; integrated variability management across disciplines; sharing architectural knowledge; PL organisations and governance; and integrating agile development and PLE.

4.2 Theme Clusters

Through discussion we (the researchers and practitioners together) then identified 7 clusters of topics (Figure 3).

- **Modularity / Modularisation** refers to the process of structuring work items in distinct modules that encapsulate specific functionalities or features.
- **PL Scoping** deals with the integrated planning of member products of a PL, involved engineering domains and reusable assets to decide where invest in reuse is economically reasonable and align plans of the different planning levels.
- **Variability Management** aims for systematic handling of differences and commonalities among PL members. It involves modelling, realising, and maintaining the variations in features that allow each product within the PL to be customised or adapted for a particular customer or market segment.
- **Process** approaches organise the lifecycle of PLs and involved activities, e.g. alignment in larger organisations, adherence with process standards, or adoption of new process models.
- **Verification and Validation (V&V)** ensures that the PL meets specified requirements and fulfils its intended purpose, through systematic testing and or systematic analyses.
- **Evolution and Maintenance** deals with the ongoing processes of updating and refining a product line to accommodate new market demands, technological advancements, and changes in customer requirements.
- **Key Performance Indicators (KPI)** measure the effectiveness and efficiency of the PLE efforts. They help assess areas like cost reduction, time to market, product quality, and customer satisfaction, providing crucial data to guide decision-making and improvements.

5 DISCUSSION AND IMPLICATIONS

We will now discuss the identified themes (clusters) in more detail – from the viewpoint of a reader who is involved in or interested in industrial application of academic research.

When collecting material and discussing research opportunities for the different clusters, we decided to merge the clusters modularity and variability management as the discussions were overlapping frequently. We discuss literature and research opportunities for the theme clusters modularity and variability management, product line scoping, verification and validation and evolution and maintenance. We kept the theme clusters process and KPI for future work. For all team clusters, we defined one responsible researcher who is an expert in the respective area and will drive further initiatives.

5.1 Modularity and Variability Management

Literature. A plethora of approaches to variability management, variability modelling as well as implementing variable systems, including approaches to modularise systems, have been proposed since the early 1990s [64]. The most prominent variability modelling approaches are feature modelling and decision modelling [19]. However, there are also many other approaches, such as OVM [62], UML-based approaches [31], and diverse textual variability languages [7]. Many variability management tools have been proposed [6], mostly, however, as academic prototypes and only few commercial[2] or well-maintained open-source tools[3]. There were several initiatives to define a standard, e.g., CVL [32] and UVL [76], the latter initiative is still ongoing and very active [4].

Diverse programming paradigms have been proposed to realise modular, highly-variable systems, e.g., feature-oriented software development [2] and delta-oriented programming [70], to name but a few. Diverse industry case studies have been published [80] and collected [51]. There are also many textbooks [2, 17, 62, 80].

There have been attempts to capture the variability modelling body of knowledge[5] as well as many surveys [6, 19, 64]. There are several online courses[6] and teaching material collections[7] available, for variability and modularisation specifically, and the overarching topic SPLE in general. Common topics related to variability management and modularisation that are taught at universities are

[2]https://www.pure-systems.com/purevariants, https://biglever.com/solution/gears/
[3]https://featureide.github.io/
[4]https://github.com/Universal-Variability-Language, https://modevar.github.io/
[5]https://github.com/SECPS/VMBoK
[6]https://github.com/SoftVarE-Group/Course-on-Software-Product-Lines
[7]http://teaching.variability.io/

object-oriented programming and design patterns for variability, clone-and-own with version control and build systems, feature-oriented programming, services and microservices, and frameworks and plugins.

Besides SPLC[8], there is a dedicated international (working) conference on variability modelling[9].

Research Opportunities. Rabiser et al. [65, 66, 73] have concluded based on multiple empirical studies, that while the perception of academics and industry practitioners is different, there is at least some agreement on the challenges of adopting software product lines. It remains also a key challenge for academics to transfer their research into industry and for industrial practitioners to get access to knowledge from our (academic) community.

The workshop which provided the inspiration and material for this paper is further proof of that, while also being a way to alleviate this challenge by bringing together academia and industry to discuss existing solutions (such as those described in this section) and problems (such as those described in the previous sections).

Regarding variability and modularisation the workshop as well as discussions with practitioners and academics at the workshop and afterwards indicate the following being important avenues for future research motivated by industry challenges:

The need to deal with *legacy systems* complicates variability management and motivates refactoring such legacy systems into a modular architecture, to even enable an SPL approach. That this is often not done results from the associated human and financial effort. Automation of this *refactoring* (into a PL [41]) process is a necessity. Specifically, approaches for variability (or feature) location [68] or mining [40] from existing systems, automated generation of reusable artefacts and variability models [4], automated support for checking and fixing inconsistencies between artefacts and models [83], automated support for variability model and engineering artefact maintenance [50] are important. While there are initial efforts in all these directions (some cited above), those typically are limited to very specific scenarios and artefacts and not yet flexible enough to be applied in practice.

The size of real-world systems is motivating research to deal with variability management and modularisation in a way that works in practice. Put differently, *scalability* of existing approaches, e.g., *for variability modelling and modularisation needs to be investigated more, in real empirical studies*, not just with toy examples [65].

Extracting variability knowledge, mostly available in unstructured documents or the heads of engineers, is not just a technical challenge and motivates further research. Particularly, *managing variability often requires interdisciplinary knowledge*, not "just" software engineering. Especially in cyber-physical and software-intensive systems contexts, other disciplines need to be involved. There are initiatives towards multi-disciplinary approaches to variability management [26, 27] to better cover this issue.

Variability management in the future might also benefit from recent *generative AI approaches*, particularly Large Language Models (LLMs), for instance, to assist in variability modelling and variability implementation [1, 30]. In form of a "co-pilot" we expect LLMs to already now be very helpful for industry practitioners.

[8]https://splc.net
[9]https://vamosconf.net/

5.2 Product Line Scoping

Literature. PL scoping defines the member products of a PL and the major (externally visible) common and variable features of the member products, analyses the products from an economic point of view, and controls and schedules the development, production, and marketing of the product line and its products [36]. This is a pivotal activity to ensure the economical success of a PL. In scoping we should take a problem-oriented perspective ("What does the customer value most?"), but we also need to consider solution-oriented aspects ("What can we implement most easily?") [75]. Various approaches have been proposed [56, 71] and the field has been surveyed [38, 44, 48] over time. Several industrial case studies and experience reports exist [15, 20, 23, 39]. Approaches such as Modular Function Deployment (MFD) [25, 29, 43] or Metus [8, 42] should be considered from an industrial perspective.

Research Opportunities. Scoping necessitates a multidisciplinary approach, and it would be beneficial to explore how various methods can be integrated across engineering disciplines, product levels, and organisations. There is a need for case studies examining the impact of scoping on modularisation decisions, particularly given the increasing influence of software on product innovation. As in many other PL areas that need to process heterogeneous input, partly consisting of natural language, PL Scoping might benefit from applying LLMs. A particular research challenge, is to start from inconsistent, incomplete information and, nevertheless, arrive at a scope representation that (1) is precise, consistent, and complete and (2) contains sufficient semantics to be useful for further activities (e.g., by defining features and constraints on their combinations, by defining included/excluded functionality in representations amenable for automatic processing).

5.3 Verification and Validation

Literature. Verification and validation in SPLE aims to ensure quality while coping with large numbers of product configurations and constant change. Efforts have to be balanced between domain and application engineering. The field has been surveyed [22, 37, 45, 58, 77] and some industrial experience reports exist [67, 80]. Testing reference configurations, pairwise testing [47, 49, 57, 60], model-based testing [34] approaches, and hardware virtualisation approaches help to mitigate V&V complexity. Several formal analysis techniques have been applied to product lines or have been extended to consider the particular properties for product lines, e.g., Type Checking, Static Analysis, Model-checking, Theorem Proving. Such analyses can then be designed as product-based analyses, family-based analyses (considering the whole PL), or feature-based analyses (looking at the implementation or dependencies of a particular feature) [78]. Somewhere in between product-based and family-based analyses are sample-based approaches that analyse/test the PL by sampling a subset of products. While they reduce complexity, they risk missing defects due to incomplete coverage [3].

Research Opportunities. An area with ongoing research and potential opportunities is traceability between requirements, variability models, and implementation. This is, e.g., needed to ensure coverage of requirements and identify gaps, e.g., to validate regulatory compliance. Such approaches should maximise "return-on-investment", by reducing cost for usage (e.g., through tool support) and increasing the value (e.g., by automatically generating and executing tests that are mapped to requirements).

5.4 Evolution and Maintenance

Literature. Like any software system that addresses the real world, a software product line has to undergo continuous adaptation and change if it is to remain satisfactory in use [46]. Compared to general software evolution, for PLs the situation gets more complex due to their specific characteristics [13]:

- *Long life-span.* A PL is a long-term investment, with increased return-on-investment with more derived products. On the other hand, it must also evolve to reflect new products.
- *Large size and complexity.* An SPL represents a whole family of systems, with multiple teams involved, distributed knowledge, and evolution of different parts happening at different speeds.
- *More interdependencies.* For instance, changes in PL level can affect many products; new requirements on product level can require changes for the whole PL.

Evolution gets even more complex in approaches that use multiple PLs [72], e.g., multiple PLs with a shared architecture [21], hierarchical PLs [12], sharing across PLs [81].

Deelstra et al. [21] and Schmid et al. [74] report on evolution strategies that commonly occur in practice: In *proactive* evolution future requirements are *planned in advance* and added on the PL level. This is usually a pure domain engineering activity, often based on business goals. In constrast, there are three common ways to deal with changes *in hindsight*, during product derivation:

- During *reactive* evolution requirements are implemented as they arise, e.g., as variable assets. Advantages are immediate availability on PL level and avoidance of product-specific implementations. Highly automated approaches often aim for this strategy so that complete products can be derived automatically. Disadvantages are required frequent changes on PL level and need to co-evolve products.
- In *branch-and-unite*, requirements are initially handled on product level, e.g., by creating new product-specific branches, which are later reunified with the SPL. This way, PL changes can be reduced and emphasis can be put on the product first. However, the merge can become complex and error-prone.
- A *bulk* situation occurs when we create too many branches by evolving on product level. This can lead to maintenance problems and major effort is required to reintegrate all branches.

There are some model-driven approaches to PL evolution, e.g., EvoFM [13, 61]. Some models focussing on differences can also be used to describe evolution, e.g., delta models by Schäfer et al. [70] or change sets by Hendrickson et al. [35]. However, the overall use of such approaches in industrial practice is limited.

Many other approaches to PL evolution have been described, well summarised in a systematic literature review by Marques et al. [50]. They conclude that there is no consensus about the evolution process and that while case studies are quite popular, few industry-sized case studies are publicly available. Furthermore, few of the proposed PL evolution techniques come with tool support.

There is some tool support for change impact analysis. For instance, Heider et al. [33] use regression testing to asses the effects of changes in variability models on derived products. This can potentially provide rapid feedback on unintended consequences of changes. However, in general tool support is sparse [50].

Research Opportunities. Within evolution of PLs, the research area with most potential for industrial practice is probably automation and tool support. This affects all areas mentioned above, e.g., identification of features, construction of variability models, and transformation of assets with variability into a full product line.

A related topic that currently receives a lot of attention, and might also be somewhat over-hyped, is whether LLMs are useful for these tasks, e.g., to refactor artefacts or to transform artefacts into a PL. For general development operations, e.g., refactoring, extracting methods, or reorganising a class hierarchy there is powerful support in IDEs like IntelliJ IDEA[10] or Microsoft Visual Studio Code[11] – this is only getting amplified by recently added AI assistant functions like GitHub copilot. Similar mechanisms could be offered in feature-aware or variability-aware variants.

In software evolution, we have to deal with inconsistencies that arise naturally (e.g., when requirements change but the implementation has not changed yet or when parts of the system change but related dependent parts have not changed yet). Handling such inconsistencies, e.g., by propagating the necessary changes through the PL could be a promising research area. Again, LLMs might be used for generating necessary artefacts/changes. However, there is little experience of how well they will perform, especially if the goal is to establish a correct and consistent PL.

We still need more empirical in-vivo research, providing, e.g., an evaluation of approaches in realistic projects, helping to understand why some projects succeed or fail in practice.

Alos, it seems that research on migration (from legacy systems into a PL) emphasises early phases (e.g., identification of features, variability analysis), with less works dealing with the transformation into a product, i.e., actually "doing" the migration [5]. Hence, there might be research opportunities in developing practical, tool-supported approaches that allow such migration.

5.5 So What?

Let us take a step back and look again at the main question this paper raises: there is considerable research in SPLE, nevertheless industry practitioners are still struggling in everyday projects when trying to make product lines work in practice – why is that?

In reviewing the workshop and the extracted information, it seems that there are *different kinds of challenges* and "pain points", which require different approaches to move forward:

- *Industry needs to catch up* – Industry does not know approaches and techniques that are actually available.

  Way forward: Industry needs to inform themselves, collaborate and communicate with researchers.

[10] https://www.jetbrains.com/idea/
[11] https://code.visualstudio.com/

- *Academics need to catch up* – They do not know enough about the real world, have biased/incomplete understanding of industrial practice.

  Way forward: They should not assume they know or rely only on literature to identify problems, but elicit problems and challenges from industry.
- *Problems are inherently hard* – Some challenges are just hard by the nature of the problem (complexity and scale of real systems, constant change caused by the need to address changing requirements and technologies).

  Way forward: Additional research needed; potentially joint forces of industry practitioners and academics can tackle the problems; technological progress opens new opportunities.

Based on these pain points we aim to explicitly answer our second research question, i.e., what are the reasons for the challenges industry has and what are existing solutions:

Answering RQ2 (Reasons and Existing Solutions)
Industry struggles to find solutions that are applicable in their domains, given their specific environment and tooling. Thus, they frequently develop their own custom solutions. Many approaches have been proposed in the literature, however, only few have been applied to real-world systems. While researchers and practitioners agree on the challenges of adopting PLE, transferring academic approaches to industry remains challenging. Inherent challenges of industry, such as legacy systems, need for automation (including variability modelling), scale and complexity of industrial systems, and the difficulty of (variability) knowledge extraction and sharing, motivate more research.

We summarise a potential way forward, from our perspective and based on our interpretation of opinions expressed by the workshop participants as answer to our third research question:

Answering RQ3 (Way Forward)
More frequent interaction of industry practitioners and researchers is required to tackle the described challenges and adapt existing approaches in diverse domains. The SPL community can also benefit from more frequent interaction to identify research opportunities and to empirically evaluate their research. Formats such as industry workshops and the SPLC industry track are appropriate to establish first contact. Afterwards, smaller groups focusing on particular theme clusters need to be created, which can meet more frequently to work on use cases and solutions.

5.6 Limitations and Threats to Validity

Generalisability. Our case companies are from a wide spectrum of domains that have traditionally participated in the SPLE community. Hence, we believe that the reported challenges and opportunities are representative for the community. On the other hand, we should be aware that there are industries with software-intensive products that have not been connected with the SPLE community, e.g., banking and financial services, cyber-security, education, robotics and autonomous vehicles. These domains might have their own objectives and challenges, which might not be reflected in the reported results. Also, companies in our case are mostly from Europe, which might also have had an influence on our results.

Selection bias. We selected companies based on our own networks. We naturally selected companies we knew are working on product lines and can report about challenges.

Influence by researchers. We actively participated in the workshop and also performed the grouping of challenges to theme clusters. However, we followed a systematic research method (thematic analysis) to extract our results.

6 RELATED WORK

Over more than three decades diverse researchers have discussed the challenges industry faced with adopting SPLs [9–11, 16, 51, 53, 80]. From the nineties to 2024, we see many challenges, especially those also described in this paper, remain relevant for industry. For example, Berger et al. [9] presented the results of a survey distributed among industry practitioners (35 responses analysed) to study to what extent/how variability modelling is done in industrial practice. Similarly to our conclusions, they conclude that industry is still experimenting with various solutions.

Berger et al. [10] also recently presented a multiple-case study in which they analyse the current adoption of variability management techniques in twelve medium- to large-scale industrial cases in domains such as automotive, aerospace or railway systems. They conclude with a brief discussion of challenges, most notably that "the adoption of SPLE concepts is still a tool-integration problem, given all the different types of artefacts and existing tooling, which engineers are familiar with and that is core to the development".

While some challenges might be hard to address due to the inherent characteristics of complex software systems, we firmly believe that cooperation between academia and industry [66] is the only way to effectively address most of the challenges.

7 CONCLUSION

Adopting system and software product line engineering remains desirable yet challenging for industry despite a plethora of approaches proposed in academia. The domain-specific needs of industry, and inherent industry challenges, such as the need to deal with legacy systems and the size and complexity of highly-variable systems, need to be taken into account.

In this paper, we summarised challenges of 9 companies from industry, clustered into 7 theme clusters. For 4 of these clusters, we discussed existing solutions to address challenges and opportunities for further research. More frequent interaction between academia and industry in diverse formats should help to exchange and discuss challenges, use cases, knowledge and solutions.

ACKNOWLEDGMENTS

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association is gratefully acknowledged. This work is supported, in part, by Science Foundation Ireland grant 13/RC/2094. We want to thank all participants of the SPLC 2023 industry challenges workshop, particularly the representatives from the nine companies who presented their challenges.

REFERENCES

[1] Mathieu Acher, José Galindo Duarte, and Jean-Marc Jézéquel. 2023. On Programming Variability with Large Language Model-based Assistant. In *Proc. of the 27th ACM International Systems and Software Product Line Conference-Volume A*. ACM, 8–14.

[2] Sven Apel, Don Batory, Christian Kästner, and Gunter Saake. 2013. Feature-oriented software product lines. (2013).

[3] Sven Apel, Alexander von Rhein, Philipp Wendler, Armin Größlinger, and Dirk Beyer. 2013. Strategies for product-line verification: case studies and experiments. In *ICSE*. 482–491.

[4] Wesley KG Assunção, Silvia R Vergilio, and Roberto E Lopez-Herrejon. 2020. Automatic extraction of product line architecture and feature models from UML class diagram variants. *Information and Software Technology* 117 (2020), 106198.

[5] Wesley K. G. Assunção, Roberto E. Lopez-Herrejon, Lukas Linsbauer, Silvia R. Vergilio, and Alexander Egyed. 2017. Reengineering legacy applications into software product lines: a systematic mapping. *Empir. Softw. Eng.* 22, 6 (2017), 2972–3016.

[6] Rabih Bashroush, Muhammad Garba, Rick Rabiser, Iris Groher, and Goetz Botterweck. 2017. Case tool support for variability management in software product lines. *ACM Computing Surveys (CSUR)* 50, 1 (2017), 1–45.

[7] Maurice H ter Beek, Klaus Schmid, and Holger Eichelberger. 2019. Textual variability modeling languages: an overview and considerations. In *Proc. of the 23rd International Systems and Software Product Line Conference-Volume B*. ACM, 151–157.

[8] Grischa Beier, Asmus Figge, Robert Müller, Uwe Rothenburg, and Rainer Stark. 2013. Supporting product development through cross-discipline dependency-modeling–novel approaches for traceability-usage. *Lecture Notes on Information Theory Vol* 1, 1 (2013).

[9] Thorsten Berger, Ralf Rublack, Divya Nair, Joanne M Atlee, Martin Becker, Krzysztof Czarnecki, and Andrzej Wąsowski. 2013. A survey of variability modeling in industrial practice. In *Proc. of the 7th International Workshop on Variability Modelling of Software-intensive Systems*. ACM, 1–8.

[10] Thorsten Berger, Jan-Philipp Steghöfer, Tewfik Ziadi, Jacques Robin, and Jabier Martinez. 2020. The state of adoption and the challenges of systematic variability management in industry. *Empir. Softw. Eng.* 25, 3 (2020), 1755–1797.

[11] Jan Bosch. 1999. Product-line architectures in industry: a case study. In *Proc. of the 21st international conference on Software engineering*. 544–554.

[12] Jan Bosch. 2000. *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach.* Addison-Wesley.

[13] Goetz Botterweck and Andreas Pleuss. 2014. Evolution of Software Product Lines. In *Evolving Software Systems*, Tom Mens, Alexander Serebrenik, and Anthony Cleve (Eds.). Springer, 265–295.

[14] Virginia Braun and Victoria Clarke. 2012. *Thematic analysis.* American Psychological Association.

[15] Marta Cecilia Camacho, Francisco Álvarez, César A. Collazos, Paul Leger, Julián Dario Bermúdez, and Julio Ariel Hurtado. 2021. A Collaborative Method for Scoping Software Product Lines: A Case Study in a Small Software Company. *Applied Sciences* 11, 15 (July 2021), 6820.

[16] Lianping Chen and Muhammad Ali Babar. 2010. Variability management in software product lines: an investigation of contemporary industrial challenges. In *International Conference on Software Product Lines*. Springer, 166–180.

[17] Paul Clements and Linda M. Northrop. 2002. *Software product lines - practices and patterns.* Addison-Wesley.

[18] Daniela S. Cruzes and Tore Dybå. 2011. Recommended Steps for Thematic Synthesis in Software Engineering. In *Proc. of the 5th International Symposium on Empirical Software Engineering and Measurement, ESEM 2011, Banff, AB, Canada, September 22-23, 2011*. IEEE Computer Society, 275–284.

[19] Krzysztof Czarnecki, Paul Grünbacher, Rick Rabiser, Klaus Schmid, and Andrzej Wąsowski. 2012. Cool features and tough decisions: a comparison of variability modeling approaches. In *Proc. of the 6th International workshop on variability modeling of software-intensive systems*. ACM, 173–182.

[20] Ivonei Freitas da Silva, Paulo Anselmo da Mota Silveira Neto, Pádraig O'Leary, Eduardo Santana de Almeida, and Silvio Romero de Lemos Meira. [n. d.]. Software product line scoping and requirements engineering in a small and medium-sized enterprise: An industrial case study. 88 ([n. d.]), 189–206.

[21] Sybren Deelstra, Marco Sinnema, and Jan Bosch. 2005. Product derivation in software product families: a case study. *JSS* 74, 2 (2005), 173–194.

[22] Ivan do Carmo Machado, John D. McGregor, Yguaratã Cerqueira Cavalcanti, and Eduardo Santana De Almeida. 2014. On strategies for testing software product lines: A systematic literature review. *Information and Software Technology* 56, 10 (2014), 1183–1199.

[23] Dominik Domis, Stephan Sehestedt, Thomas Gamer, Markus Aleksy, and Heiko Koziolek. 2014. Customizing domain analysis for assessing the reuse potential of industrial software systems: experience report. In *Proc. of the 18th International Software Product Line Conference - Volume 1*. ACM, Florence Italy, 310–319.

[24] Yael Dubinsky, Julia Rubin, Thorsten Berger, Slawomir Duszynski, Martin Becker, and Krzysztof Czarnecki. 2013. An Exploratory Study of Cloning in Industrial Software Product Lines. In *Proc. of CSMR 2013*. 25–34.

[25] Gunnar Erixon. 1998. *Modular function deployment.* https://swepub.kb.se/showrecord?q=onr%3A%22swepub%3Aoai%3ADiVA.org%3Akth-2619%22&n=1&d=swepub&noredirect=true&tab2=absamp%3Blanguage%3Den

[26] Hafiyyan Sayyid Fadhlillah, Antonio M Gutiérrez Fernández, Rick Rabiser, and Alois Zoitl. 2023. Managing Cyber-Physical Production Systems Variability using V4rdiac: Industrial Experiences. In *Proc. of the 27th ACM International Systems and Software Product Line Conference-Volume A*. ACM, 223–233.

[27] Miao Fang, Georg Leyh, Joerg Doerr, and Christoph Elsner. 2016. Multi-variability modeling and realization for software derivation in industrial automation management. In *Proc. of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*. 2–12.

[28] FeatureIDE. 2024. *FeatureIDE.* https://featureide.github.io Accessed on 2024-04-24.

[29] Antonio Wagner Forti, César Coutinho Ramos, and Jorge Muniz Jr. 2023. Integration of design structure matrix and modular function deployment for mass customization and product modularization: a case study on heavy vehicles. *The International Journal of Advanced Manufacturing Technology* 125, 3 (2023), 1987–2002. Publisher: Springer.

[30] José A Galindo, Antonio J Dominguez, Jules White, and David Benavides. 2023. Large Language Models to generate meaningful feature model instances. In *Proc. of the 27th ACM International Systems and Software Product Line Conference-Volume A*. ACM, 15–26.

[31] Hassan Gomaa. 2005. Designing software product lines with UML. In *29th Annual IEEE/NASA Software Engineering Workshop-Tutorial Notes (SEW'05)*. IEEE, 160–216.

[32] Øystein Haugen, Andrzej Wąsowski, and Krzysztof Czarnecki. 2012. CVL: common variability language. In *Proc. of the 16th International Software Product Line Conference-Volume 2*. ACM, 266–267.

[33] Wolfgang Heider, Rick Rabiser, Paul Grünbacher, and Daniela Lettner. 2012. Using regression testing to analyze the impact of changes to variability models on products. In *SPLC*. 196–205.

[34] Christopher Henard, Mike Papadakis, Gilles Perrouin, Jacques Klein, and Yves Le Traon. 2013. Assessing Software Product Line Testing Via Model-Based Mutation: An Application to Similarity Testing. In *Sixth IEEE International Conference on Software Testing, Verification and Validation, ICST 2013 Workshops Proceedings, Luxembourg, Luxembourg, March 18-22, 2013*. IEEE Computer Society, 188–197.

[35] Scott A. Hendrickson and Andre van der Hoek. 2007. Modeling Product Line Architectures through Change Sets and Relationships. In *ICSE*. 189–198.

[36] ISO 26550:2015(E) 2015. *Software And Systems Engineering - Reference Model For Product Line Engineering And Management.* Standard. International Organization for Standardization, Geneva, CH.

[37] Martin Fagereng Johansen, Øystein Haugen, and Franck Fleurey. 2011. A Survey of Empirics of Strategies for Software Product Line Testing. In *Fourth IEEE International Conference on Software Testing, Verification and Validation, ICST 2012, Berlin, Germany, 21-25 March, 2011, Workshop Proceedings*. IEEE Computer Society, 266–269.

[38] Isabel John and Michael Eisenbarth. [n. d.]. A decade of scoping: a survey. In *Proc. of the 13th International Software Product Line Conference*. 31–40.

[39] Isabel John, Jens Knodel, Theresa Lehner, and Dirk Muthig. [n. d.]. A practical guide to product line scoping. In *10th International Software Product Line Conference (SPLC'06)*. IEEE, 3–12.

[40] Christian Kästner, Alexander Dreiling, and Klaus Ostermann. 2013. Variability mining: Consistent semi-automatic detection of product-line features. *IEEE Transactions on Software Engineering* 40, 1 (2013), 67–82.

[41] Miguel A Laguna and Yania Crespo. 2013. A systematic mapping study on software product line evolution: From legacy system reengineering to product line refactoring. *Science of Computer Programming* 78, 8 (2013), 1010–1034.

[42] Thorsten Lammers, Matthias Guertler, and Henning Skirde. 2022. Can product modularization approaches help address challenges in technical project portfolio management?–Laying the foundations for a methodology transfer. *International Journal of Information Systems and Project Management* 10, 2 (2022), 26–42.

[43] Mark W. Lange and Andrea Imsdahl. 2014. Modular Function Deployment: Using Module Drivers to Impart Strategies to a Product Architecture. In *Advances in Product Family and Product Platform Design*. New York, NY, 91–118.

[44] Jihyun Lee, Sungwon Kang, and Danhyung Lee. 2010. A Comparison of Software Product Line Scoping Approaches. *Int. J. Softw. Eng. Knowl. Eng.* 20, 5 (2010), 637–663.

[45] Jihyun Lee, Sungwon Kang, and Danhyung Lee. 2012. A survey on software product line testing. In *Proc. of the 16th International Software Product Line Conference - Volume 1*. ACM, Salvador Brazil, 31–40.

[46] Meir Lehman and Juan C. Fernáandez-Ramil. 2006. *Software Evolution.* Wiley, Chapter 1, 7–40.

[47] Malte Lochau, Sebastian Oster, Ursula Goltz, and Andy Schürr. 2012. Model-based pairwise testing for feature interaction coverage in software product line engineering. *Softw. Qual. J.* 20, 3-4 (2012), 567–604.

[48] Luciano Marchezan, Elder Rodrigues, Wesley Klewerton Guez Assunção, Maicon Bernardino, Fábio Paulo Basso, and João Carbonell. 2022. Software product line scoping: A systematic literature review. *Journal of Systems and Software* 186

(April 2022), 111189.
[49] Dusica Marijan, Arnaud Gotlieb, Sagar Sen, and Aymeric Hervieu. 2013. Practical pairwise testing for software product lines. In *Proc. of the 17th International Software Product Line Conference*. ACM, Tokyo Japan, 227–235.
[50] Maíra Marques, Jocelyn Simmonds, Pedro O Rossel, and María Cecilia Bastarrica. 2019. Software product line evolution: A systematic literature review. *Information and Software Technology* 105 (2019), 190–208.
[51] Jabier Martinez, Wesley KG Assunção, and Tewfik Ziadi. 2017. Espla: A catalog of extractive spl adoption case studies. In *Proc. of the 21st International Systems and Software Product Line Conference-Volume B*. ACM, 38–41.
[52] Jabier Martinez, Wesley K. Assunção, and Tewfik Ziadi. 2017. *ESPLA: A Catalog of Extractive SPL Adoption Case Studies*. https://but4reuse.github.io/espla_catalog/ Accessed on 2024-04-27.
[53] Andreas Metzger and Klaus Pohl. 2014. Software product line engineering and variability management: achievements and challenges. *Future of software engineering proceedings* (2014), 70–84.
[54] Miro. 2024. *Miro online whiteboard*. https://miro.com/online-whiteboard/ Accessed 2024-04-24.
[55] Mukelabai Mukelabai, Damir Nešić, Salome Maro, Thorsten Berger, and Jan-Philipp Steghöfer. 2018. Tackling combinatorial explosion: a study of industrial needs and practices for analyzing highly configurable systems. In *Proc. of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. 155–166.
[56] Muhammad Asim Noor, Rick Rabiser, and Paul Grünbacher. [n. d.]. A collaborative approach for reengineering-based product line scoping. In *1st International Workshop on Agile Product Line Engineering (APLE'06), Maryland, USA*.
[57] Sebastian Oster, Florian Markert, and Philipp Ritter. 2010. Automated Incremental Pairwise Testing of Software Product Lines. In *Software Product Lines: Going Beyond*. Vol. 6287. 196–210. Series Title: Lecture Notes in Computer Science.
[58] Sebastian Oster, Andreas Wübbeke, Gregor Engels, and Andy Schürr. 2011. A Survey of Model-Based Software Product Lines Testing.
[59] David Lorge Parnas. 1976. On the Design and Development of Program Families. *IEEE Trans. Software Eng.* 2, 1 (1976), 1–9.
[60] Gilles Perrouin, Sebastian Oster, Sagar Sen, Jacques Klein, Benoit Baudry, and Yves Le Traon. 2012. Pairwise testing for software product lines: comparison of two approaches. *Softw. Qual. J.* 20, 3-4 (2012), 605–643.
[61] Andreas Pleuss, Goetz Botterweck, Deepak Dhungana, Andreas Polzer, and Stefan Kowalewski. 2012. Model-driven Support for Product Line Evolution on Feature Level. *Journal of Systems and Software* 85, 10 (October 2012), 2261–2274.
[62] Klaus Pohl, Günter Böckle, and Frank Van Der Linden. 2005. *Software product line engineering: foundations, principles, and techniques*. Vol. 1. Springer.
[63] pure systems. 2024. *pure::variants*. https://www.pure-systems.com/purevariants Accessed on 2024-04-24.
[64] Mikko Raatikainen, Juha Tiihonen, and Tomi Männistö. 2019. Software product lines and variability modeling: A tertiary study. *Journal of Systems and Software* 149 (2019), 485–510.
[65] Rick Rabiser, Klaus Schmid, Martin Becker, Goetz Botterweck, Matthias Galster, Iris Groher, and Danny Weyns. 2018. A study and comparison of industrial vs. academic software product line research published at SPLC. In *Proc. of the 22nd International Systems and Software Product Line Conference*. ACM, 14–24.
[66] Rick Rabiser, Klaus Schmid, Martin Becker, Goetz Botterweck, Matthias Galster, Iris Groher, and Danny Weyns. 2019. Industrial and academic software product line research at SPLC: perceptions of the community. In *Proc. of the 23rd International Systems and Software Product Line Conference*. ACM, 27:1–27:6.
[67] Sacha Reis, Andreas Metzger, and Klaus Pohl. 2007. Integration Testing in Software Product Line Engineering: A Model-Based Technique. Lecture Notes in Computer Science, Vol. 4422. 321–335.
[68] Julia Rubin and Marsha Chechik. 2013. A survey of feature location techniques. *Domain Engineering: Product Lines, Languages, and Conceptual Models* (2013), 29–58.
[69] Julia Rubin, Andrei Kirshin, Goetz Botterweck, and Marsha Chechik. 2012. Managing forked product variants. In *Proc. of SPLC 2012*. 156–160.
[70] Ina Schaefer, Lorenzo Bettini, Viviana Bono, Ferruccio Damiani, and Nico Tanzarella. 2010. Delta-oriented programming of software product lines. In *Software Product Lines: Going Beyond: 14th International Conference, SPLC 2010, Jeju Island, South Korea, September 13-17, 2010. Proceedings 14*. Springer, 77–91.
[71] Klaus Schmid. 2002. A comprehensive product line scoping approach and its validation. In *Proc. of the 24th International Conference on Software Engineering, ICSE 2002, 19-25 May 2002, Orlando, Florida, USA*. ACM, 593–603.
[72] K. Schmid and H. Eichelberger. 2008. A requirements-based taxonomy of software product line evolution. *Electronic Communications of the EASST* 8 (2008).
[73] Klaus Schmid, Rick Rabiser, Martin Becker, Goetz Botterweck, Matthias Galster, Iris Groher, and Danny Weyns. 2021. Bridging the gap: voices from industry and research on industrial relevance of SPLC. In *Proc. of the 25th International Systems and Software Product Line Conference*. ACM, 184–189.
[74] K. Schmid and M. Verlage. 2002. The economic impact of product line adoption and evolution. *IEEE Software* 19, 4 (2002), 50–57.
[75] Daniel Simon and Thomas Eisenbarth. 2002. Evolutionary Introduction of Software Product Lines. In *SPLC*. 272–283.
[76] Chico Sundermann, Kevin Feichtinger, Dominik Engelhardt, Rick Rabiser, and Thomas Thüm. 2021. Yet another textual variability language? a community effort towards a unified language. In *Proc. of the 25th ACM International Systems and Software Product Line Conference-Volume A*. ACM, 136–147.
[77] Antti Tevanlinna, Juha Taina, and Raine Kauppinen. 2004. Product family testing: a survey. *ACM SIGSOFT Software Engineering Notes* 29, 2 (March 2004), 12–12.
[78] Thomas Thüm, Sven Apel, Christian Kästner, Ina Schaefer, and Gunter Saake. 2014. A Classification and Survey of Analysis Strategies for Software Product Lines. *ACM Comput. Surv.* 47, 1 (2014), 6:1–6:45.
[79] Mojtaba Vaismoradi, Hannele Turunen, and Terese Bondas. 2013. Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & health sciences* 15, 3 (2013), 398–405.
[80] Frank van der Linden, Klaus Schmid, and Eelco Rommes. 2007. *Software product lines in action - the best industrial practice in product line engineering*. Springer.
[81] Rob van Ommering. 2005. Software reuse in product populations. *IEEE Trans. Soft. Eng.* 31, 7 (july 2005), 537–550.
[82] Verband der Automobilindustrie e.V., (VDA). 2024. Automotive SPICE. https://vda-qmc.de/en/automotive-spice/. Accessed: 2024-04-08.
[83] Michael Vierhauser, Paul Grünbacher, Alexander Egyed, Rick Rabiser, and Wolfgang Heider. 2010. Flexible and scalable consistency checking on product line variability models. In *Proc. of the 25th IEEE/ACM international conference on automated software engineering*. 63–72.