Proceedings of the Design Society, Volume 6: DESIGN 2026
https://doi.org/10.1017/pds.2026.10616

AI-supported variant management activities – insights from an industrial case study

Fionn Winger[1]✉, Benedikt Müller[1], Daniel Roth[1], Bruno Barciela Diaz-Blanco, Fatos Elezi and Matthias Kreimeyer[1]

[1] *Institute for Engineering Design and Industrial Design, University of Stuttgart, Germany*

✉ fionn.winger@iktd.uni-stuttgart.de

ABSTRACT: Variant management faces increasing complexity that challenges traditional rule-based configuration approaches. This contribution explores how AI can support the generation of configuration rules (1) by comparing two solution concepts – a deterministic Python-based and an LLM-based approach. Following a structured early-stage AI system development methodology, the research investigates (2) how AI can be methodically integrated into variant management and (3) how implementation factors differ between both approaches. The results reveal distinct trade-offs between transparency and efficiency.

KEYWORDS: variant management, artificial intelligence (AI), case study, configuration management, large language model (LLM)

1. Introduction

Industrial product development (PD) is increasingly characterized by rising product diversity and data intensity, leading to growing challenges in variant and complexity management (Krause & Gebhardt, 2018). To address differentiated customer demands, companies must continuously expand their product portfolios while maintaining cost efficiency, quality, and development speed (Seiler & Krause, 2020). The simultaneous growth of data volumes, variant interdependencies, and configuration rules (CR) creates significant complexity, making efficient variant management a strategic necessity (Krause & Gebhardt, 2018). However, as product architectures become more modular and digitalized, variant management processes, particularly those related to configuration, documentation, and the creation and maintenance of CR, remain largely manual and knowledge-intensive and expert-driven (Deschner, 2020; Frey et al., 2023). Consequently, the costs associated with managing and adapting variant portfolios increase substantially, while the potential for errors and inefficiencies grows in parallel (Krause & Gebhardt, 2018).

Recent advancements in artificial intelligence (AI) offer new opportunities to support or even automate such rule-based and data-intensive activities (Cooper, 2024). AI methods enable the recognition of complex patterns and relationships in heterogeneous data, potentially facilitating automated generation, validation, and maintenance of CR (Pathirannehelage, 2025). Nevertheless, despite significant industrial interest and investment, many AI initiatives fail to deliver measurable value due to missing process integration, insufficient contextualization, and inadequate stakeholder involvement (Challapally et al., 2025). Successful AI implementation in engineering domains therefore requires structured, transparent, and interdisciplinary approaches that ensure alignment between domain objectives, available data, and technical feasibility (Romeo & Lacko, 2025; Müller et al., 2025a). Clearly articulated problem statements, realistic data requirements, and expected outcomes enable organizations to anticipate barriers such as insufficient data quality or integration challenges before they impede model development and validation (Lee et al., 2023; Romeo & Lacko, 2025). In this regard, collaborative early-stage activities - such as ideation, elaboration, and assessment of domain-specific AI use cases - play a crucial role (Müller et al., 2025a). Structured approaches, including the use of a variety of

https://doi.org/10.1017/pds.2026.10616 Published online by Cambridge University Press

existing methods, enhance communication between stakeholders, support traceability of decisions, and increase trust in AI systems (Micheli et al., 2023; Königstorfer, 2024; Müller et al., 2025a). However, empirical knowledge concerning how these structured, early-stage methodologies affect interdisciplinary collaboration and AI system development outcomes in industrial PD remains scarce. In particular, little is known about how the joint involvement of domain and AI experts influences the identification, selection, and implementation of AI use cases in complex domains such as variant management.

1.1. Objectives and research question

Against this background, the present study explores the potential of AI to support variant management, focusing specifically on the generation and revision of configuration rules. Conducted as an industrial case study within a large manufacturing company, the research applies a systematic interdisciplinary methodology for AI system development (Müller et al., 2025a) using dedicated method cards (Müller et al., 2025b–d) to guide the early stages of domain-specific AI system development. The objective of this research is twofold: (1) to identify, assess, and prioritize AI use cases within variant management, and subsequently implement one selected use case up to a Technology Readiness Level (TRL) of 4, thereby demonstrating functional maturity under laboratory conditions; and (2) to empirically examine the interplay between structured methodological approaches and interdisciplinary collaboration. Accordingly, the study is guided by the following research questions: **RQ1**: ***How can AI contribute to the generation and revision of configuration rules in variant management?*** **RQ2**: ***How does a structured methodological approach influence the planning and execution of AI system development in variant management and what impact does the early involvement of domain experts have on the relevance and domain alignment of resulting AI solutions?***

The paper contributes to both research and practice by (i) providing empirical insights into AI integration in variant management, (ii) evaluating a structured, method-driven approach for early-stage AI system development, and (iii) deriving implications for future industrial AI adoption and collaboration between domain experts and AI experts. The remainder of this contribution is structured as follows: Section 2 reviews the theoretical background and related work in variant management and AI use case development, while Section 3 presents the methodology applied in the case study. Section 4 discusses the results, subsequently Section 5 outlining implications, limitations, and future research directions.

2. Theoretical application domain background

Variant management uses three strategies to manage product portfolios (Krause & Gebhardt, 2018). Variant reduction eliminates variants to reduce costs, variant avoidance prevents superfluous variants by allowing only the technically necessary, variant control manages existing diversity (Krause & Gebhardt, 2018; Deschner, 2020). Variant control can be defined as a systematic approach to managing growing product diversity and reducing internal complexity (Krause & Gebhardt, 2018). The establishment and maintenance of configuration rules ensure that each customer configuration is uniquely assigned to a technical solution, including the bill of materials (Frey et al., 2023). This forms the basis for automated and correct configurations in highly diverse product portfolios (Frey et al., 2023). In the context of evolving product portfolios, it is essential to ensure the continuous maintenance of rules and configuration logic to maintain consistency (McKay et al., 2021). Otherwise, inconsistencies can lead to critical production errors, and such dynamic maintenance is a fundamental prerequisite for achieving efficiency and competitiveness (Frey et al., 2023; McKay et al., 2021).

Figure 1 shows the process in the product architecture department. Each step's required data and stakeholders are shown as symbols. In a multitude of domains, variant management is predominantly executed manually (Burkhart et al., 2020). Despite the availability of tools designed to facilitate the overview of datasets, human analysts are still responsible for deriving conclusions from these sets (Frey et al., 2023). These conclusions are then used as a basis for subsequent measures. Moreover, a substantial amount of data from various domains and tools is necessary to manually adjust configuration rules (Frey et al., 2023; Deschner, 2020). In addition to the substantial and opaque datasets, this increases the complexity in the domain of product architecture.

https://doi.org/10.1017/pds.2026.10616 Published online by Cambridge University Press

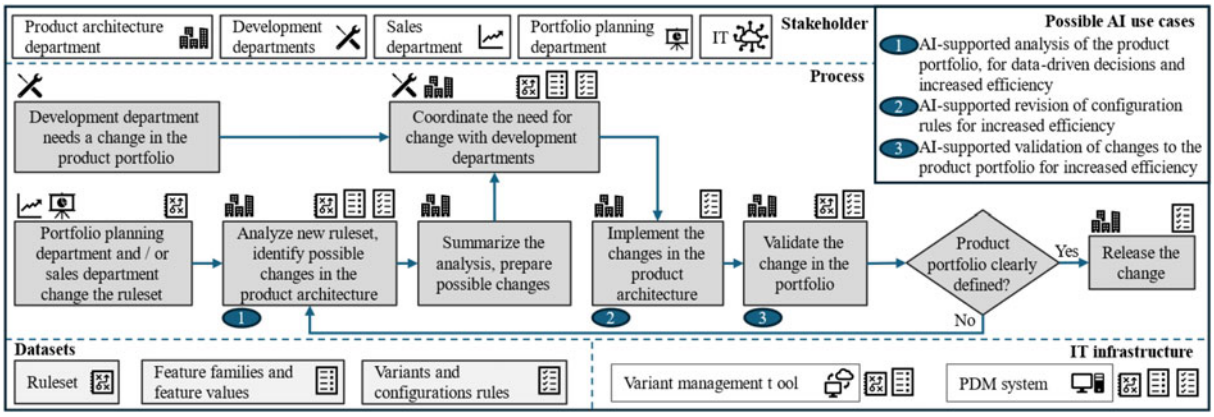


Figure 1. Overview of application domain processes, stakeholder, data, and IT infrastructure

3. Research design and methodology

The research design of this study follows a qualitative, exploratory-evaluative approach aimed at empirically examining the feasibility and applicability of a structured methodological framework for identifying and developing AI use cases in the context of variant management. The following sections outline the case study design (Section 3.1), involved participants (Section 3.2), applied methodological framework (Section 3.3), and data collection and analysis procedure (Section 3.4).

3.1. Case study design

This research adopts a single-case study design conducted in an industrial environment to empirically investigate the applicability of a methodical approach for identifying and evaluating AI use cases in variant management. The company in this case study is a large enterprise with international operations in the commercial vehicle industry, characterized by variant-rich product portfolios and highly configurable products that can be tailored by end customers. The investigated application domain focuses on the documentation and maintenance of product architectures, which currently rely on extensive manual coordination and suffer from fragmented data and IT structures (cf. Section 2.1). Multiple heterogeneous IT systems and the absence of data continuity result in high data dynamics and significant coordination efforts across departments. The study has an exploratory and evaluative character (TRL 1-4) and was conducted between March and September 2025. Its objective is twofold: first, to empirically examine the feasibility, benefits, and challenges of AI-generated configuration rules, and second, to evaluate the usefulness and transparency of a structured methodological procedure and associated method cards (Müller et al., 2025a-d) that support interdisciplinary teams in identifying, documenting, and assessing AI opportunities in complex industrial contexts. The case selection rationale is based on the representativeness of variant-intensive product lines and the company's strategic interest in exploring AI-driven approaches to manage configuration complexity. The project involves multiple stakeholders (cf. Section 3.2) and follows an iterative, interdisciplinary process (cf. Figure 2), reflecting the collaborative nature and real-world challenges of industrial AI system development.

3.2. Participants

The case study involved a multidisciplinary team representing the key stakeholder groups relevant to AI system development in industrial contexts (see Figure 2). A total of seven participants were engaged, covering complementary areas of expertise essential for variant management and AI integration. The group of domain experts (n = 3) consisted of experienced product architects with in-depth knowledge of domain-specific processes, data structures, and IT tools. They represented different organizational levels – one team leader (DTL) and two specialists (D1 & D2) – ensuring a balanced perspective on both strategic and operational activities within product architecture management (cf. stakeholder in Figure 1). An interface role (IF, n = 1) was held by a product architect with dual expertise in the domain and AI methods. As the Card Owner, this participant mediated between domain experts, IT specialists, and AI professionals, ensuring continuous alignment throughout all study stages. The IT expert (IT, n = 1) contributed insights into the company's IT infrastructure, system interfaces, and data maintenance processes, supporting the assessment of data accessibility and technical feasibility for AI solutions

https://doi.org/10.1017/pds.2026.10616 Published online by Cambridge University Press

(cf. stakeholder in Figure 1). Finally, two AI experts (AI1 & AI2, n = 2) from the company's Center of Excellence for AI (CoE-AI) provided methodological and technical expertise in AI system development. Although initially unfamiliar with the applied method cards, they supported data interpretation, feasibility evaluation, and reflection on methodological transferability to other industrial use cases. The participants' expertise was represented across all stages (see Figure 2), reflecting an iterative and collaborative process that integrated both domain relevance and methodological rigor.

3.3. Methodological framework

The methodological framework applied in the case study follows the AI system development approach by Müller et al. (2025a), which operationalizes the early stages of AI projects through transparent and traceable procedures. The framework comprises the stages of preparation, ideation, assessment, prioritization, and execution, thereby linking strategic planning with technical implementation. The overall procedure, participant interactions, and data collection activities are illustrated in Figure 2.

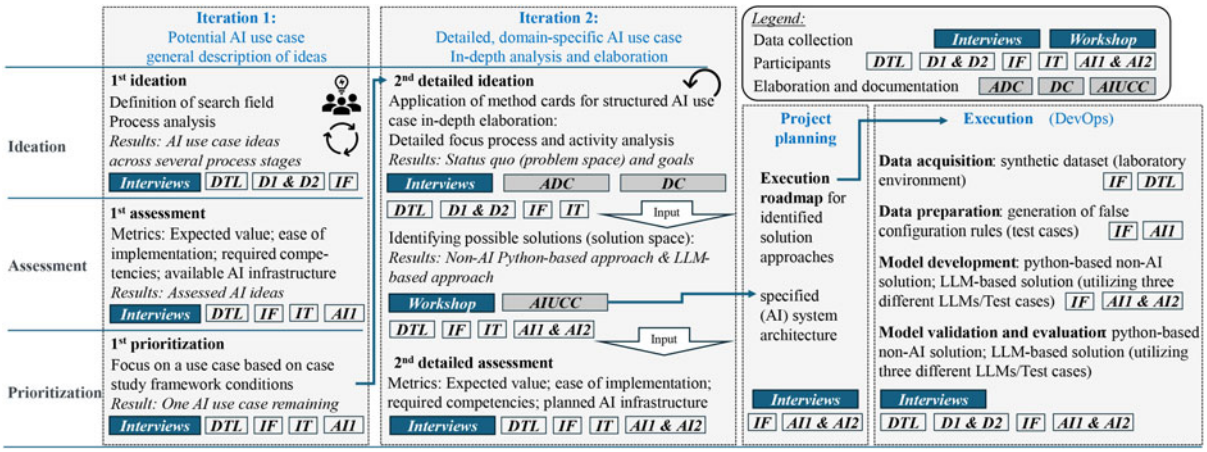


Figure 2. Case study procedure and data collection

The first four stages constitute the conceptual front end and form the focus of this study. In the **Preparation** stage, organizations define AI-related focus areas and assess their project conditions. The **Ideation** stage aims to identify potential AI use cases by concentrating on the respective problem, data, and technology. During **Assessment**, identified cases are evaluated with regard to feasibility, expected benefits, and resource requirements. This is followed by **Prioritization**, in which a selection of use cases is aligned with strategic objectives and prepared for execution. The subsequent **Execution** stage marks the transition to technical realization using existing technology-oriented iterative AI system development processes such as MLOps or CRISP-ML. To operationalize these early stages, the study applied the method card set developed by Müller et al. (2025b–d): The **Application Domain Card (ADC)** captures the domain context, key processes, challenges, and stakeholders, forming the problem-oriented foundation (status quo) for AI opportunity identification. The **Data Card (DC)** - with a data-oriented perspective - documents the status quo regarding available data sources, ownership, quality, and accessibility along the data life cycle, ensuring a structured link to international standards (e.g. ISO/IEC 8183:2023). The **AI Use Case Card (AIUCC)** - with a technology-oriented perspective – integrates both status quo perspectives as a basis for deriving the envisioned AI functionality, system boundaries, input-output relations, and key requirements for implementation. Based on the results (problem space and framework conditions for the solution space), potential AI solutions are identified, developed further, and detailed into a domain-specific AI use case concept.

The framework was applied in two iterative ideation cycles. In the first iteration, potential AI use cases within variant management were identified and assessed on a general level, leading to the selection of the use case "AI-generated configuration rules" as the most promising candidate. The second iteration focused on a detailed elaboration of this use case using all method cards.

3.4. Data collection and analysis

Data collection and analysis in this case study followed a mixed-method approach. Multiple data sources - including semi-structured interviews, a workshop, and project documentation - were systematically

https://doi.org/10.1017/pds.2026.10616 Published online by Cambridge University Press

integrated and analyzed within the framework of the methodological card set (cf. Section 3.3). This triangulation enhanced the reliability and internal validity of the findings. The respective data collection activities and their temporal relation to the case study process are illustrated in Figure 2. **Interviews** were conducted by the IF (Card Owner) with domain experts (DTL, D1, D2), the IT specialist, and the AI experts (AI1, AI2) to gather all relevant information required for developing the ADC and DC. Due to the iterative nature of the study, interviews were held at multiple points in time – either individually or in small, topic-related groups–and included regular validation sessions to ensure the accuracy and completeness of the documented content. Building on the results of the ADC and DC, a 2.5-hour **workshop** was conducted within Ideation 2 (participants: IF, DTL, AI1, and AI2 (IT for validation and feasibility check afterwards)). The workshop aimed to identify and evaluate potential AI-based solution concepts based on the previously documented problem and data characteristics. It consisted of three phases: (1) a short alignment (≈ 30 min) to review goals and context, (2) an ideation (≈ 45 min) to generate possible AI solutions through expert input and literature consultation, and (3) an evaluation and detailed elaboration (≈ 75 min) focusing on the assessment and detailing of identified solution approaches. The AIUCC served as the central elaboration support and documentation artifact during this workshop, consolidating all results into a structured and traceable format. Data analysis was conducted as qualitative content analysis – including transcription, coding, and interpretation of interview and workshop material. From the interviews, thematic categories were coded based on method card content, then documented and validated to ensure consistency and transparency across all stages of the study.

4. Case study execution and results

The following section describes the execution of the case study, based on the methodological approach outlined in Chapter 3 (cf. Figure 2). The results achieved and decisions made are presented below.

4.1. Iteration 1

The first iteration of the study focused on identifying and evaluating potential AI use cases within the department of technical product architecture planning. During **Ideation 1**, a comprehensive process analysis was conducted within the defined search field to map all relevant process steps, iterations, decision paths, stakeholders, required data, and associated IT systems. The resulting process model, visualized as a flow diagram (see Figure 1), revealed fragmented data structures, multiple system interfaces, and significant coordination effort between departments. Based on this analysis, three potential AI use cases were identified through targeted interviews with domain experts:

1. AI-supported **analysis of the product portfolio** - aimed at supporting domain experts in analyzing complex configuration data structures to enable data-driven decision-making.
2. AI-supported **revision of configuration rules** - designed to assist the creation and correction of CR by accelerating the retrieval of valid feature values and checking syntax consistency.
3. AI-supported **validation of product portfolio changes** - intended to simulate and visualize the effects of planned modifications before their implementation in the PLM system.

The structured visualization of processes, data dependencies, and involved tools proved highly valuable for aligning understanding across all stakeholders. Beyond the AI-focused objectives, the developed artifacts were perceived as beneficial within the department, serving as internal documentation and onboarding material for rarely executed or complex processes. However, the ideation stage also demonstrated that the identification of domain-specific AI opportunities is time-intensive and requires iterative expert involvement to ensure contextual accuracy and relevance. During **Assessment 1**, the identified use case ideas were recorded and discussed in a semi-structured manner. The evaluation, jointly conducted by the Card Owner (IF), DTL, IT, and one AI expert (AI1), assessed each use case based on domain-specific feasibility, potential impact, and organizational prerequisites. The assessment revealed that the three use cases build on one another (cf. Figure 1), with the configuration rule revision representing a logical and technically feasible starting point for further exploration. In **Prioritization 1**, the assessment results were consolidated, and the use case "AI-supported revision and generation of configuration rules" was selected for detailed elaboration in the subsequent iteration. The selection was based on its manageable complexity, high practical relevance, and potential to generate early insights for later, more advanced AI initiatives. The remaining use cases were documented and retained in a backlog for future development. These activities concluded the first iteration, providing a validated and shared understanding of key processes, challenges, and data structures within product architecture planning.

https://doi.org/10.1017/pds.2026.10616 Published online by Cambridge University Press

Building on these insights, the project proceeded directly to Ideation 2, focusing on the detailed elaboration and conceptualization of the selected use case.

4.2. Iteration 2

The second iteration (Ideation 2) focused on the detailed elaboration of the prioritized use case "AI-supported revision and generation of configuration rules." Building on the results of Ideation 1, a refined process and activity analysis was conducted using the ADC and DC to capture process steps, data dependencies, and quality aspects in greater detail. Iterative validation sessions with domain experts ensured the accuracy and consistency of the documentation. The consolidated activity sequence - including inputs, tasks, and outputs - is illustrated in Figure 3.

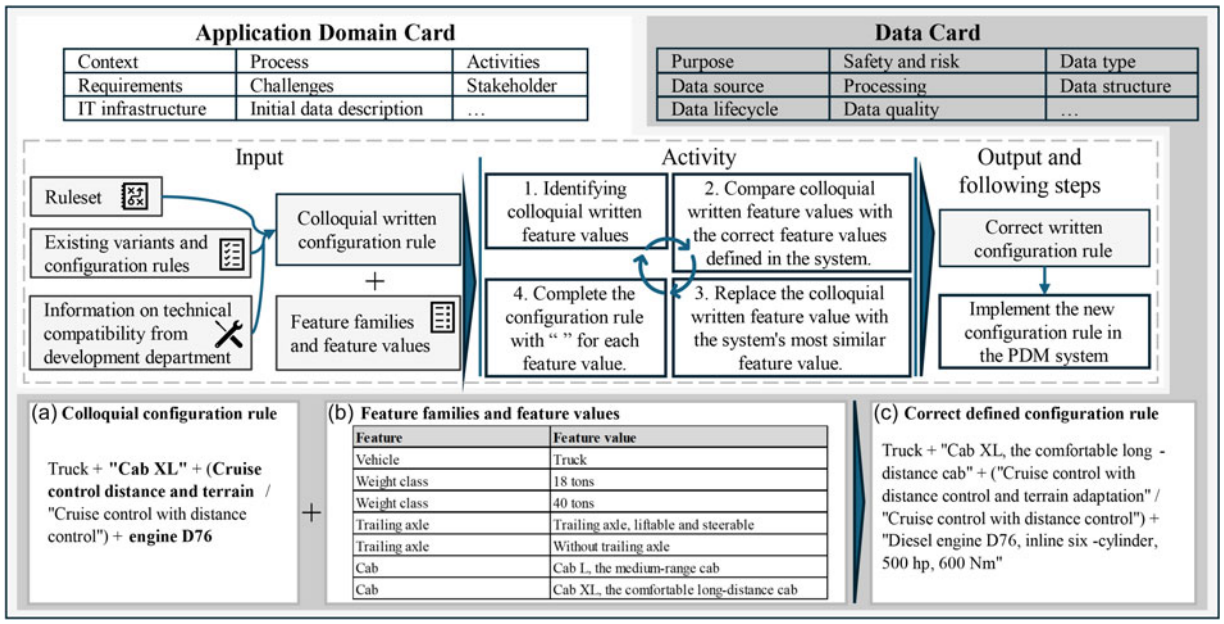


Figure 3. Input, activity, and output of the AI use case, revision of configuration rules, and example representation of a) Colloquial configuration rule, b) Feature families and feature values, c) Correct defined configuration rule

The analysis revealed that the core challenge in the focus activity lies in correctly matching colloquial configuration rules with more than 15,000 feature values stored in the company's database. Each feature value must appear in the rule exactly as registered in the system to ensure proper compilation within the product data management (PDM) environment. Based on these insights, a workshop involving the Card Owner and AI experts was conducted to identify potential solution approaches and define corresponding AI tasks. Using the AIUCC, two alternative solutions were developed:

- **Non-AI Python-based approach** - a deterministic algorithm comparing colloquial configuration rules with feature families and values, automatically replacing incorrect strings with system-compliant ones.
- **LLM-based approach** - prompt-driven use of a large language model already available in the company, designed to interpret and correct colloquial configuration rules by aligning them with predefined feature families and values.

Figure 4 provides a conceptual and technical description of the two approaches, outlining data flow, user and developer perspectives, and integration requirements. A synthetic dataset replicating the original industrial data was created to enable comparative tests without exposing sensitive information. To ensure objective evaluation, a set of domain-specific evaluation criteria was defined and validated with domain experts. These criteria assess (C1) preservation of rule semantics, (C2) correctness of feature value replacement, (C3) number of iterations required to obtain a valid rule, and (C4) transparency in handling and replacing feature values when there are multiple correct defined feature values with similar semantics. Together, these criteria form the basis for assessing performance, feasibility, and traceability of the proposed approaches in subsequent evaluation phases.

https://doi.org/10.1017/pds.2026.10616 Published online by Cambridge University Press

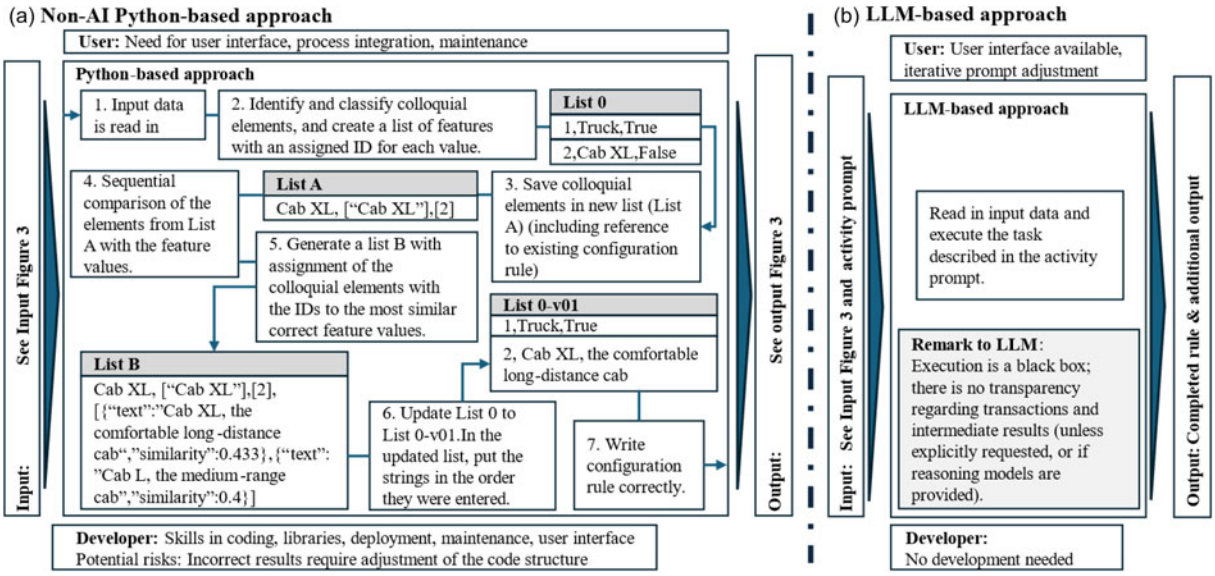


Figure 4. Identified solution approaches: a) Non-AI Python-based approach, b) LLM-based approach

4.3. Execution

The development and evaluation phase aimed to operationalize and validate the two identified solution approaches: a non-AI Python-based prototype and three large language model (LLM)-based variants. The system design was defined during the Ideation 2 workshop (cf. Figure 4), establishing functional boundaries, data flow, and evaluation criteria. To enable reproducible testing, a synthetic dataset was created to mirror the structure and logic of the company's industrial configuration data while excluding proprietary information. Three conceptual configuration rules – representing short, medium, and long variants – were designed, each containing colloquial feature values to be replaced by correct, system-compliant ones (cf. Figure 3 a & b). The Python-based solution was implemented using the libraries re, difflib.SequenceMatcher, json, pandas, and typing. Development and testing were conducted in the Atom environment and Windows command prompt. The algorithm performed deterministic text matching and replacement between colloquial configuration rules and the database of 1,000 feature values. Intermediate transformation steps were output to ensure full transparency and traceability of each operation. The prototype reached TRL 3 after initial functionality testing and TRL 4 following successful validation with the synthetic dataset. For the LLM-based solutions, three LLMs were evaluated under identical test conditions: ChatGPT 5.0, ChatGPT 4o, and Llama 3.2-vision:11b. A unified activity prompt described the task, provided the list of feature families and values, and included the colloquial configuration rule as input. Manual iterative adjustments were made to optimize prompt clarity and output consistency. The LLMs were tested within the same environment and with the same dataset to ensure comparability. All solutions were qualitatively evaluated by domain experts (DTL, D1 & D2, IF) using the predefined criteria C1, C2, C3, C4 (cf. Section 4.2): The evaluation results, which were assessed from application domain experts, are summarized in Table 1. The Python-based approach achieved consistent performance across all three test cases, completing each configuration rule correction in a single system run while maintaining full traceability. Both ChatGPT 5.0 and ChatGPT 4o successfully replaced all feature values and preserved rule semantics but required multiple iterations, particularly for longer configuration rules. In contrast, Llama 3.2-vision:11b exhibited notable inconsistencies: In the short rule, a feature value was replaced with an incorrect but similar entry, while the omission of one feature value in the long rule altered the rule's meaning. After five unsuccessful attempts, the experiment with this model was discontinued. Among the evaluated LLMs, ChatGPT 5.0 was the only one to demonstrate traceability for the medium-length CR by explicitly establishing the exchange of feature values. This involved the reasoning used in this query, with the individual steps performed and the respective drop-down menus and the creation of a semantic similarity ranking and an explanation of the rationale behind selecting the most analogous feature value. For all other CR, ChatGPT 5.0 and the remaining LLMs produced outputs without providing transparency into their decision-making process.

https://doi.org/10.1017/pds.2026.10616 Published online by Cambridge University Press

Table 1. Evaluation of the python-based and LLM-based approach variants

Criteria	Python Based	Chat GPT 5.0	Chat GPT 4o	Llama3.2-vision:11b
C1 - Semantic correctness	Y / Y / Y	Y / Y / Y	Y / Y / Y	N / Y / N
C2 - Correct replacement	Y / Y / Y	Y / Y / Y	Y / Y / Y	Y / Y / Y
C3 - Iterations	1* / 1* / 1*	2 / 1 / 1	2 / 3 / 4	3 / 3 / 5+ **
C4 - Transparency	Y / Y / Y	N / Y / N	N / N / N	N / N / N
Legend: Y = Yes, N = No // * = One-time system run // ** = Termination of the experiment				

Overall, the evaluation demonstrated that deterministic, rule-based approaches offer greater transparency and reliability, whereas LLM-based solutions provide higher automation potential but require further refinement to ensure robustness and semantic consistency in domain-specific contexts.

4.4. Discussion of solution approaches

The results presented in Section 4.3 demonstrate that both approaches are representative of a fundamental solution and fulfill the task. Table 2 presents a comparative analysis of the two approaches with respect to various implementation factors. The comparison of the two approaches was carried out on the basis of the workshop and its application to the case study described in this contribution.

Table 2. Comparing non-AI Python-based approach and LLM-based approach

	Non-AI Python-based approach	LLM-based approach
Implementation effort	High effort due to development, maintenance, testing, adaptation, and deployment.	Low effort, LLM is available with no development necessary, intuitive use, only prompt instructions required.
Scalability	Availability (databases, access, updates, support, stability, compatibility) and training are challenging.	Everyone has access to LLM, the prompt and user guide require minimal training.
Transparency	Interim results are comprehensible and transparent.	No transparency in execution, only in the result - and even then only if requested or if reasoning models are used.
Adaptability	Complex: The code must adapt when data or steps change.	Low-effort customization and exclusive input prompt customization.
Resource demands	Person with coding, database, hardware, maintenance, and development skills.	No resources other than the user are required, LLM infrastructure is available.

The Python approach is more complex and requires a significant investment in development, including programming, testing, integration, and additional measures such as maintenance, adaptation, and quality assurance. This is typical for TRL 3-4 implementations. The Python approach was developed in three cycles, with one developer and the map manager in each cycle. The LLM approach uses existing infrastructures and models, so independent development or testing is not needed. This approach requires only input prompts with data and instructions, reducing the time and qualification effort.
The Python approach offers greater control but requires its own infrastructure. Setup, maintenance, and expansion are costly in terms of technology and personnel. Additionally, scalability must be considered in the planning phase. The LLM approach is inherently scalable due to the vendor-provided infrastructure and standardization, as large computing capacities are available without additional development effort. However, data protection guidelines and the reproducibility of input prompts must be taken into account.
A further distinction is evident in the level of transparency exhibited. The Python approach is more comprehensible because it reveals the intermediate steps (cf. Figure 4), making validation and troubleshooting easier. With the LLM approach, transparency is limited, internal decision-making processes remain hidden, and only the final results are delivered. Traceability is only possible if requested explicitly via the prompt. Additionally, the LLMs Llama3.2-vision:11b and GPT 4o showed an increasing error rate as the configuration rules grew longer, reducing confidence in task performance.

https://doi.org/10.1017/pds.2026.10616 Published online by Cambridge University Press

The Python approach is limited in terms of adaptability because changes to data structures, logic, or formats require manual code adjustments by technical specialists. The LLM approach is more flexible because adjustments are made via prompts, and it supports various data types. This enables rapid changes, even by users, but comes at the expense of reproducibility and stability.
Furthermore, the approaches differ in terms of resource demands. The Python-based approach necessitates the expertise of qualified specialists in programming and integration, as well as dedicated resources for development, testing, and maintenance. However, these competencies frequently fall outside the typical skill set of most individuals. The LLM approach mitigates the technical challenges associated with the implementation of new infrastructure. However, new organizational challenges emerge concerning governance, consistency, and accountability.
In summary, the Python-based approach demonstrated superiority in terms of transparency, stability, and traceability. Conversely, the LLM-based approach exhibited accelerated deployment and enhanced accessibility. In industrial environments, a hybrid approach that integrates deterministic validation (Python) with generative support (LLM) emerges as the optimal strategy to ensure the reliability and scalability of AI-assisted configuration rule management.

5. Implications for research and industry

The findings of this study corroborate the efficacy of the method cards as structured instruments in the early phase of AI exploration. The contributions of this study are twofold: They deepen the understanding of AI integration into variant management and provide a basis for expanding the discourse on hybrid approaches in complex, rule-based domains. Furthermore, the study underscores the significance of transparency and traceability in AI systems, particularly with regard to the discordance between deterministic and generative methods. A detailed analysis of existing processes and data is imperative for any AI initiative, as it provides insights for domain optimization. The study's findings provide actionable guidance for initiating AI projects in complex domains and establish a framework for hybrid teams comprising domain and AI specialists. The method cards have been demonstrated to facilitate interdisciplinary communication and promote domain-specific AI system development. The study also demonstrates that not all use cases can be solved solely with AI. Thorough preparation enables open, technology-neutral solution identification and underscores the complementary strengths of non-AI and AI-based methods. These aspects offer novel insights for integrating hybrid systems and balance automation, transparency, and traceability in industrial AI.

6. Conclusion and future work

This case study explored the potential of AI to support the revision of configuration rules in variant management and validated a methodological approach for early AI system development. The results indicate that AI-based approaches are suitable but differ in robustness and transparency. Deterministic methods provide traceability and stability, while generative approaches offer faster implementation and flexibility. Combining both methods proves promising for industrial applications. In accordance with the aforementioned findings, the two research questions that guided this study addressed the following:

- Core contribution: structured integration of AI and non-AI approaches for variant management.
- Closing reflection: systematic frameworks drive industrial AI maturity and lasting adoption.
- The role of structured methodologies (cards) in guiding domain expert-driven ideation.

To answer RQ1, artificial intelligence (AI) can automate the process of revising colloquially formulated configuration rules by assigning colloquial feature values to predefined feature values. In the study, this objective was accomplished through the implementation of both LLM prompts and a deterministic Python-based approach. Two of the three LLMs provided accurate replacements; however, multiple iterations were necessary in some cases and the process lacked transparency. To answer RQ2, the method card-based approach structured the problem, data, and solution space, improving interdisciplinary communication and ensuring traceable decisions. Early involvement of domain experts guaranteed relevance and data realism, reducing project risks and accelerating decisions. This approach increased domain alignment of AI solutions while making limitations and next steps transparent.
The applied approach enabled a detailed and structured understanding of the problem domain and feasible AI solutions, but certain limitations must be acknowledged. Access to internal company data and documentation was restricted due to confidentiality agreements, limiting the publication of detailed

https://doi.org/10.1017/pds.2026.10616 Published online by Cambridge University Press

process descriptions and raw data. All sensitive or business-critical information was anonymized or replaced by representative examples to preserve confidentiality and data protection. The study focuses on a single industrial case, so the findings are not generalizable, but the methodological transparency ensures reproducibility in comparable industrial contexts.

Future research should address the limitations of this single-case study and validate the results in other domains and with real datasets. The results are currently at TRL 4, but efforts should explore higher maturity levels. Automated data preparation, stress tests with more data, and generalization are needed, along with research into hybrid approaches that combine deterministic validation with generative AI for scalability and transparency in industrial applications.

References

Burkhart, J., *et al.* (2020). A digital modelling approach for design configuration and manufacturing exemplified by a self-balancing scooter. *Procedia CIRP*, 91, 214–219. Elsevier. https://doi.org/10.1016/j.procir.2020.02.170

Challapally, A., *et al.* (2025). The GenAI Divide STATE OF AI IN BUSINESS 2025 [Preliminary Findings from AI Implementation Research from Project NANDA]. MIT NANDA. https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf

Cooper, R. G. (2024). The Artificial Intelligence Revolution in New-Product Development. *IEEE Engineering Management Review*, 52(1), 195–211. https://doi.org/10.1109/EMR.2023.3336834

Deschner, C. (2020). Enhanced model-based engineering for centrally managed configuration management in product lifecycle management. SHS Web of Conferences, 77, Article 03002. EDP Sciences. https://doi.org/10.1051/shsconf/20207703002

Frey, A., *et al.* (2023). Creation and validation of systems for product and process configuration based on data analysis. *Production Engineering*, 17, 263–277. https://doi.org/10.1007/s11740-022-01176-1

International Organization for Standardization. (2023). *Information technology - Artificial intelligence - Data life cycle framework (ISO/IEC 8183:2023)*. ISO.

Königstorfer, F. (2024). A comprehensive review of techniques for documenting artificial intelligence. *Digital Policy, Regulation and Governance*, 26(5), 545–559. https://doi.org/10.1108/DPRG-01-2024-0008

Krause, D., & Gebhardt, N. (2018). *Methodische Entwicklung modularer Produktfamilien: Hohe Produktvielfalt beherrschbar entwickeln*. Springer Vieweg. https://doi.org/10.1007/978-3-662-53040-5

Lee, M. C. M., *et al.* (2023). The implementation of artificial intelligence in organizations: A systematic literature review. *Information & Management*, 60(5), 103816. https://doi.org/10.1016/j.im.2023.103816

McKay, A., *et al.* (2021). Maintaining consistency across design descriptions in engineering product development. In C. McMahon (Ed.), *Proceedings of the Design Society, Volume 1: ICED21* (pp. 1987–1996). Cambridge University Press. https://doi.org/10.1017/pds.2021.460

Micheli, M., *et al.* (2023). The landscape of data and AI documentation approaches in the European policy context. *Ethics and Information Technology*, 25(4), 56. https://doi.org/10.1007/s10676-023-09725-7

Müller, B., Roth, D., & Kreimeyer, M. (2025a). KI-Adaption in der Produktentwicklung: Ansätze und Vorgehen aus wissenschaftlicher und praktischer Perspektive. *Zeitschrift Für Wirtschaftlichen Fabrikbetrieb*, 120(s1), 44–50. https://doi.org/10.1515/zwf-2024-0158

Müller, B., Roth, D., & Kreimeyer, M. (2025b). From theory to practice: derivation of the application domain card for supporting AI system development. *Proceedings of the Design Society*, 5, 1903–1912. https://doi.org/10.1017/pds.2025.10204

Müller, B., Roth, D., & Kreimeyer, M. (2025c). Democratized AI System Development for SMEs: The Data Card as a Practical Method. In K. Hölzle *et al.* (Eds.), *Stuttgarter Symposium für Produktentwicklung SSP 2025* (pp. 340–351). Fraunhofer IAO. https://doi.org/ https://doi.org/10.18419/opus-16366

Müller, B., Roth, D., & Kreimeyer, M. (2025d). AI System Development for SMEs: The AI Use Case Card as a Practical Method. In K. Hölzle *et al.* (Eds.), *Stuttgarter Symposium für Produktentwicklung SSP 2025* (pp. 376–387). Fraunhofer IAO. https://doi.org/ https://doi.org/10.18419/opus-16366

Pathirannehelage, S. H., *et al.* (2025). Design principles for artificial intelligence-augmented decision making: An action design research study. *European Journal of Information Systems*, 34(2), 207–229. https://doi.org/10.1080/0960085X.2024.2330402

Romeo, E., & Lacko, J. (2025). Adoption and integration of AI in organizations: a systematic review of challenges and drivers towards future directions of research. Kybernetes. https://doi.org/10.1108/K-07-2024-2002

Seiler, F. M., & Krause, D. (2020). A multi-dimensional configuration algorithm for modular product architectures. Proceedings of the International Design Conference – DESIGN 2020, 2405-2414. Cambridge University Press. https://doi.org/10.1017/dsd.2020.283

https://doi.org/10.1017/pds.2026.10616 Published online by Cambridge University Press