

Article

Planning Product Upgrades: A Method for Defining Release Types and Their Strategies for Software-Intensive Products

Armin Stein * , Umut Volkan Kizgin , Mohammad Albittar and Thomas Vietor 

Institute for Engineering Design, Technische Universität Braunschweig, Hermann-Blenk Strasse 42,
38108 Braunschweig, Germany

* Correspondence: armin.stein@tu-braunschweig.de

Abstract

The environment of today's companies is marked by increasing dynamism. Rapid technological developments, strong innovation impulses, and continual market entry of new competitors create volatile conditions that make the delivery of valuable products challenging. Long-term corporate success therefore depends on offering a product portfolio consistently aligned with evolving market needs. Customers expect products that show continuous improvements in performance and functionality over time, making systematic product upgrading a key success factor. Release planning addresses this need by enabling continuous product evolution through planned product upgrades. It focuses on selecting and combining functional units for structured publication within releases. This proactive management of product value offers substantial potential but also demands comprehensive know-how, particularly given rising product complexity and the interplay of multiple technologies. The objective of this work is to develop a methodology that supports effective planning of product upgrades. The method assists in the product-specific selection of release types and the derivation of suitable release strategies. It yields release units defined by product structure and provides recommendations for appropriate release strategies. The methodology is demonstrated through its application to an electric vehicle, illustrating its practical relevance for software-intensive products.

Keywords: release planning; product upgrades; release types; release strategies; modular product architecture; software-intensive products; product complexity management; lifecycle-oriented product development



Academic Editor: Cheng-Fu Yang

Received: 16 December 2025

Revised: 23 January 2026

Accepted: 26 January 2026

Published: 28 January 2026

Copyright: © 2026 by the authors.

Published by MDPI on behalf of the International Institute of Knowledge Innovation and Invention. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

1.1. Motivation

Maintaining or increasing a product's value through targeted product evolution is a key success factor for companies. To achieve this, products are enhanced with additional performance and excitement features that are introduced to the market as individual releases [1]. The shift from purely mechatronic products to technical systems creates new opportunities for planning such upgrade units [2]. Software-based components enable upgrades during product use, opening new possibilities regarding frequency, organization, and execution. Release planning addresses this challenge through systematic, proactive product evolution [3], yet requires defining upgrade units and publication strategies [4]. Due to product complexity, methodological support is essential for selecting release types and deriving suitable release strategies. From a scientific perspective, however, current solutions remain insufficient. Existing approaches predominantly focus on either purely

physical or purely digital domains, lacking a cohesive framework that derives release types directly from the modular product structure of hybrid systems. Consequently, there is a lack of established methods to systematically translate component complexity into differentiated release strategies for software-intensive products.

Market competition constitutes an additional central driver of product update decisions because it increases the opportunity costs of delayed upgrades and can shorten the effective decision window for introducing improvements. Consequently, decisions on update scope and cadence are frequently shaped not only by internal capabilities but also by anticipated competitor moves and the risk of competitive disadvantage [5,6].

1.2. Businesses in Dynamic Markets

Today's companies face numerous challenges that make successful product development and marketing increasingly demanding. Globalization and the shift from seller to buyer markets create heterogeneous demand patterns requiring adequate corporate responses [7,8]. Rapidly emerging and changing markets demand high responsiveness to evolving customer needs, making continuous product adaptation a vital competence [9]. Rising product complexity, shortening life cycles [10], and the need for standardization to reduce costs [11] reinforce the importance of a competitive product strategy balancing customer orientation and economic success [10]. Multi-domain products further require extensive internal know-how and organizational flexibility [12].

1.3. Customer Benefit and Satisfaction as the Key to Business Success

Customers confront companies with rising expectations, making individual need satisfaction increasingly relevant. Over time, more product functions are assumed without higher willingness to pay [13], reinforcing the importance of focusing on true customer value [13]. Customer value results from perceived benefit relative to perceived price: Product value = perceived benefit/perceived price [14]. Companies must therefore maximize benefit while balancing internal costs and considering whether customers acknowledge performance increases [7]. Customer benefit differs from product benefit: the latter stems from technical functionality, while the former depends on contextual usage [15]. For instance, an electric vehicle may offer 400 km product range but only 350 km customer range at 30 °C with air conditioning [16]. Customer satisfaction emerges when expectations align with actual performance [17]. According to the Kano model, satisfaction builds on basic, performance, and excitement requirements [16–18], highlighting that exceeding expectations is essential for differentiation [19].

1.4. Products Through the Ages

Market developments directly affect products, which now exhibit increasing variety and growing functional scope due to rising customer individuality and competitive pressure [4]. Although product interpretations vary, literature typically distinguishes between supplier and customer perspectives. From Brockhoff's view, a product is an allocation of properties designed to satisfy diverse present and future market demands [20]. From a company perspective, it represents the output of combined production factors, including tangible and intangible goods [21]. In this work, the focus lies on physical products that [9] meet needs as technical goods.

Heterogeneous customer requirements drive the evolution from products to technical systems [22]. Increasingly, products integrate software, hardware, and electronics, enabling intelligent mechanical units and new functionalities [23]. Porter et al. describe four stages of intelligent systems, monitoring, control, optimization, autonomy, each increasing customer value [24]. The automotive industry exemplifies this shift, with software complexity growing to more than ten million lines of code and accounting for over 40% of production

costs [25]. Consequently, companies require strong cross-domain expertise and cost-efficient strategies to align system capabilities with customer value [2,9].

Shortening product lifecycles further demand continuous market adaptation [9]. The economic product lifecycle describes phases from market entry to decline [26], where strategic gaps signal the need for upgrades and innovations [27]. Technology lifecycles must also be considered, as different components, particularly software, follow distinct temporal patterns [28–30].

Finally, growing product complexity, both internal and external, creates significant challenges for development and stakeholder alignment [1,8]. Balancing these dimensions is essential, and release management has emerged as a suitable means to manage market and product dynamics effectively [31].

1.5. From the Product Structure to the Product Architecture

Strategic orientation alone is insufficient; strategic goals must be translated into the product substance, making product structure essential [8]. Product structure, defined in DIN 199 as a representation of components and their interrelations [32], consists of components, assemblies, and parts and their interfaces, which may be mechanical, physical or non-contact [8]. Complementing this, the functional structure maps required functions, components and dependencies, enabling allocation of customer needs to functional units [8,18]. Functional abstraction, such as distinguishing main and secondary functions of a lawn mower, supports solution exploration and customer-focused design [26].

The combination of functional and physical structure forms the product architecture, created through iterative transformation of functions during development [33]. Ulrich distinguishes modular architectures—clear function–component mappings with decoupled, interface-defined modules—from integral ones with tightly interwoven functions [34]. Architectures can be classified by functional and physical independence [26].

Because 70–80% of product costs are set in development, architectural decisions strongly influence cost, variety and competitiveness, including time-to-market and customization trade-offs [26,35]. Inkermann et al. summarize these effects into four main architectural target fields [36].

Modularization operationalizes an efficiency-oriented architecture. A module is an independent unit formed to reduce complexity [33], functionally and geometrically bounded yet interoperable via standardized interfaces [37]. Modularity reduces product and process complexity while enabling customer-focused variety, though translating market requirements into module partitions remains challenging [38]. Its key properties include decoupling, commonality, combinability, interface standardization and function binding [39]. Module drivers—motivations for modularizing components—span the entire lifecycle [18,26] and were synthesized by Erixon et al. [40]. Conflicts such as commonality vs. variety create three overarching issues of modularization: variety vs. uniformity, organizational coordination, and after-sales [41]. Reliable application requires sufficiently detailed technical functions [40], particularly for future product evolution and customer perception.

Modular thinking also supports organizational design: traditional hierarchies risk weak process and market orientation [42]. Modular organizational units with decentralized responsibility process coherent task packages, enhancing flexibility [39,43].

Finally, modularization must align with product strategy, which defines target segments, competitive approaches and the marketing mix [15,44]. Strategic goals are strongly influenced by modular structure [45]. Decoupled modules enable integrating innovative functions while reducing planning and implementation effort, making modular architecture a key strategic instrument [39]. This structural decoupling provides the necessary

theoretical foundation for release planning, as it enables the definition of independent upgrade units (releases) required for continuous lifecycle management.

1.6. Specialties of Software Intensive Products

With rising product complexity and modularization, markets show a clear shift toward digital-first products, those designed with software embedded from the outset rather than added later [46]. In industries such as automotive, software increasingly determines functionality and purchase decisions [47]. Developing software-intensive products requires considering software-specific environments: low marginal costs, high flexibility, but the need to balance customer value with resource efficiency [46]. Customers expect continuous functionality evolution rather than static software. Consequently, challenges arise regarding low-cost replication, ease of updates but high coordination effort, numerous stakeholders, and higher upgrade frequency than for physical products [48]. Software lifecycles are evolutionary, shaped by ongoing fixes and feature extensions; thus agility and short time-to-market are essential for early customer validation [46]. Software-intensive products demand an organization-level framework, for solving safety and security relevant issues [49,50]. Organizations must align traditional processes with software-driven demands, motivating the subsequent focus on release planning.

1.7. Life Cycle-Oriented Product Enhancement Through Release Planning

Sustaining customer appeal requires a product offering that consistently reflects market needs. As established in earlier chapters, upgrade measures help maintain long-term product attractiveness, and their proactive organization is addressed through release planning [51]. The central goal is to select product features based on defined criteria, bundle them, and publish them as release units for a specific product [52]. Thus, release planning is essential for managing sustainable product value aligned with market and corporate requirements [18].

The term “release” originates from software engineering, where it describes a system version meeting customer expectations [53]. In this context, a release is a planned bundle of functional enhancements published at defined intervals [54]. Releases consist of features, value-creating units integrated into the product [55]. Mandatory features enable product identification; optional features generate additional value [1]. Release planning prioritizes features, coordinates stakeholder interests, technical constraints, and resource capacities [56], ultimately enabling efficient lifecycle evolution [17].

Releases may be classified as major or minor. Major releases provide high customer impact and require strategic alignment; harmonization of module development cycles can reduce cost and resource demand [8], with major releases serving as upgrade milestones [18]. Minor releases address smaller improvements or fixes with lower customer impact and can occur independently of major cycles [4]. Their frequency depends on component change probability, especially in software-intensive products [4].

Release timing can be time-based—fixed intervals common in high-tech modular products [57], or feature-based, focusing on completing interdependent features [58]. Releases may be periodic or event-driven; choosing cycle lengths requires evaluating module flexibility, as physical modules evolve slowly while software components allow shorter cycles [8].

Different combinations of release types and timing form release strategies [4]. Automotive firms typically use low-frequency predefined cycles; smartphone industries apply higher-frequency, periodic releases. Strategy selection depends on internal factors (e.g., system flexibility), external factors (e.g., change likelihood), and alignment with product and corporate strategy. Competitor strategies may serve as benchmarks [4,59]. In markets

characterized by short imitation cycles, competitor announcements and established update rhythms act as external constraints that influence whether time-based publication or accelerated event-driven releases are preferable [4,5]. Competitor monitoring therefore represents a relevant contextual input for release strategy selection, even when the methodological core is primarily derived from product structure and complexity [6]. In markets characterized by short imitation cycles, competitor announcements and established update rhythms act as external constraints that influence whether time-based publication or accelerated event-driven releases are preferable [4,5]. Competitor monitoring therefore represents a relevant contextual input for release strategy selection, even when the methodological core is primarily derived from product structure and complexity [6].

2. State of the Art

2.1. Requirements for the Method to Be Developed

Developing a method for selecting product- and organization-specific release types and strategies requires a clearly defined design framework based on explicit requirement criteria. Following Sudhoff, this involves both general requirements that apply independently of the task and specific requirements tied directly to the objective of this work [60]. General requirements include complete consistency, reproducibility of results, user-friendliness and practical applicability, achieved through clear formulations, an unambiguous underlying system and purpose-oriented procedural steps that ensure formally and content-wise reliable outcomes [18,61]. Building on this foundation, the content-related requirements arise from the conditions of contemporary product upgrades and demand that the methodology support all planning stages, address increasing product complexity, incorporate and formalize modular product structures, enable the definition of release types, generate suitable release strategies, reflect product-specific characteristics within release planning, integrate all structural factors of release plans and continuously account for market dynamics and lifecycle developments.

2.2. Approaches to Describing Complex Product Structures

Planning releases requires a deep understanding of product structures and dependencies, making granular product knowledge essential. Relevant approaches therefore focus on modular product structures and their interactions. Braun et al. propose a matrix-based modeling method to describe complex mechatronic systems, addressing increasing integration of electronics and software and enabling interdisciplinary analysis through combined functional and component views [62]. Their Multi Domain Matrix links function and system hierarchies, allows intra- and inter-domain examination via DSM and DMM, and increases transparency of structural impacts on processes, though it is insufficient as a standalone basis for a full method. Complementing structural views, feature models capture product variability by representing mandatory and optional features hierarchically [63], following Kang's notation [64]. To incorporate market dynamics, Botterweck et al. introduce evolutionary feature models distinguishing stable and time-dependent features and mapping expected modifications along roadmaps [65]. While highly intuitive and useful for release planning, feature models lack structural linkage, making their combination with product-structure approaches necessary.

2.3. Approaches to Planning and Managing Modular Products

This chapter presents literature relevant to planning and managing complex products, beginning with Modular Function Deployment (MFD), followed by Göpfert's integrated concept of modular product development, and concluding with the Time-to-Market (TTM) management approach by Nippa et al. Erixon's MFD provides a systematic five-step

method for modularizing products by deriving customer needs through Quality Function Deployment, translating them into functional solutions, identifying modularization potential via Modultreiber using the Module Indication Matrix with 9/3/1 weighting, evaluating technical and economic feasibility, and optimizing modules accordingly [40]. While user-friendly and transparent, MFD neglects temporal changes of modules, limiting its applicability for release planning. Göpfert extends modularization by integrating technical and organizational perspectives through the METUS procedure, a five-phase iterative process that defines the problem, generates architectural alternatives, evaluates them through utility analysis, derives modular organizational units, and performs final assessments [33]. Although offering valuable insight into technical–organizational interplay, the approach focuses on early development stages and lacks consideration of market and lifecycle dynamics essential for release-type and strategy definition. Nippa et al. contribute with a TTM management framework aimed at determining optimal innovation launch timings by combining Market-Pull and Technology-Push thinking within long-, mid- and short-term planning horizons supported by innovation management and roadmapping [5]. Long-term planning evaluates market and technology impulses using scenario techniques and technology foresight; mid-term planning differentiates and synchronizes product and technology roadmaps, incorporates competitive analysis and results in an updated innovation roadmap; short-term planning translates innovations into concrete projects, aligning them with resources to derive exact market introduction dates [5,66]. Although not providing release-type classification or strategy derivation, the TTM framework offers essential temporal and strategic perspectives that inform the methodology for defining release types and strategies.

2.4. Approaches to Release Planning

This chapter presents relevant release-planning approaches, distinguishing between mechatronic and software-oriented methods to evaluate their contribution to defining release types, release strategies, and the consideration of soft- and hardware characteristics. Only approaches offering an applicable process are examined. Maurer et al. propose a cycle-oriented methodology for designing sustainably modifiable modular product structures, enabling systematic upgrades aligned with dynamic customer needs [11]. Their method identifies feature-level change drivers through expert workshops, assigns features to influencing factors, and derives an *Änderungsprioritätszahl* (change priority number) that reveals upgrade frequencies and critical upgrade times, later transferred to a product roadmap for anticipative planning [37]. While valuable for determining release timing, the approach lacks support for defining release types and focuses primarily on hardware, thereby neglecting the faster iteration cycles required for software components. Kühn's systematic method for release planning in intelligent technical systems structures strategic, tactical and operational planning [4]. Strategically, it defines release types and release strategies based on characteristic planning factors and classifies strategies as module-oriented, product-oriented or flexible, derived via a portfolio evaluation of change probabilities and system complexity [4]. Although offering important elements for strategy selection and linking product properties with planning decisions, the approach does not sufficiently support product-specific release-type derivation or timing and relies heavily on mechatronic assumptions. Sahin introduces a value-oriented approach for continuous product evolution through proactive multi-release planning [67]. His method first derives release types from characteristic attributes and domain-specific change behaviors, then assigns suitable strategies, flexible, static, resource-oriented or highly systematic, via a dimensional classification and portfolio logic. This represents the most comprehensive support for the guiding questions of this chapter, integrating both hardware and software aspects. However, Sahin's

release-type definition is strongly customer-driven and lacks deeper product-structure consideration, limiting product specificity despite strong methodological guidance.

2.5. Evaluation of Relevant Methods

The preceding chapters introduced approaches for describing complex product structures, managing modular products and supporting release planning, which must be assessed against the content requirements. Using Harvey balls, as shown in Figure 1, the evaluation shows that structural modeling methods offer valuable insight into modularity and evolutionary market dynamics but do not link to product upgrades. Approaches for managing modular architectures similarly emphasize structural representation and organizational handling without enabling transfer to release planning. In contrast, release-planning methods provide more relevant support; however, deeper detail is lacking, and only Şahin sufficiently addresses the definition of release types and strategies. Crucially, existing methods often homogenize the upgrade cycles of hardware and software, failing to provide differentiated strategies for components with distinctly different lifecycles and change frequencies in cross-domain products. Consequently, two main challenges persist: integrating modular architectures and achieving product-specific release planning that accounts for these varying dynamics. This gap motivates the development of a unified, practical and transparent methodology.

	Braun et al. [62]	Kang et al. [64]	Erixon [40]	Göpfert [33]	Nippa et al. [5]	Maurer et al. [37]	Kühn [4]	Şahin [67]
Consideration of modular product structures								
Formalization of modular product structures								
Consideration of market dynamics								
Reference to product complexity								
Typification of releases								
Alternative release strategies								
Consideration of the specifics of the release plan								
Connection of product specifics and release planning								

Legend

not fulfilled
 barely fulfilled
 partially fulfilled
 almost fulfilled
 fully fulfilled

Figure 1. Assessment of the state of the art [4,5,33,37,40,62,64,67].

3. Method for Defining Release Types and Their Strategies

3.1. Objective and Scope of Action of the Method to Be Developed Method

The methodology aims to support continuous product evolution through strategically planned improvements, grounded in the requirements and the research gap. It provides an operational tool for selecting release types and deriving release strategies, incorporating market needs, customer expectations and development factors of increasingly software-intensive products. By combining qualitative and quantitative criteria, the method ensures

transparent, traceable decisions and applicability to multi-domain products integrating hardware, software, electronics and electrical systems.

3.1.1. Procedure for Developing the Method

The methodology is developed through a four-step scheme, as shown in Figure 2, beginning with the definition of a systematic process for selecting release types and strategies. Suitable methodological tools are then identified and adapted, resulting in problem-oriented process methods. These are implemented in Microsoft Excel to create an applicable tool demonstrated using an electric vehicle, whose use also enables evaluation of the methodology.

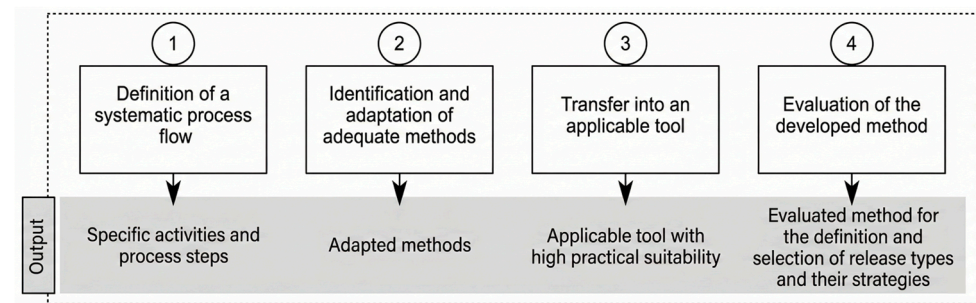


Figure 2. Procedure for developing the method.

3.1.2. Reference Process for Defining Release Types and Their Strategies

Various approaches support release planning [1,4,18,67]. Şahin addresses the definition of release types and strategies, making his process a reference framework for the methodology developed here. The focus lies on the initial step, which structures complex products into distinct release units and, based on identifiable release-type characteristics, derives suitable publication strategies [67]. Şahin's method, as shown in Figure 3, provides a valuable foundation that must be refined, particularly regarding product-specific selection of release types.

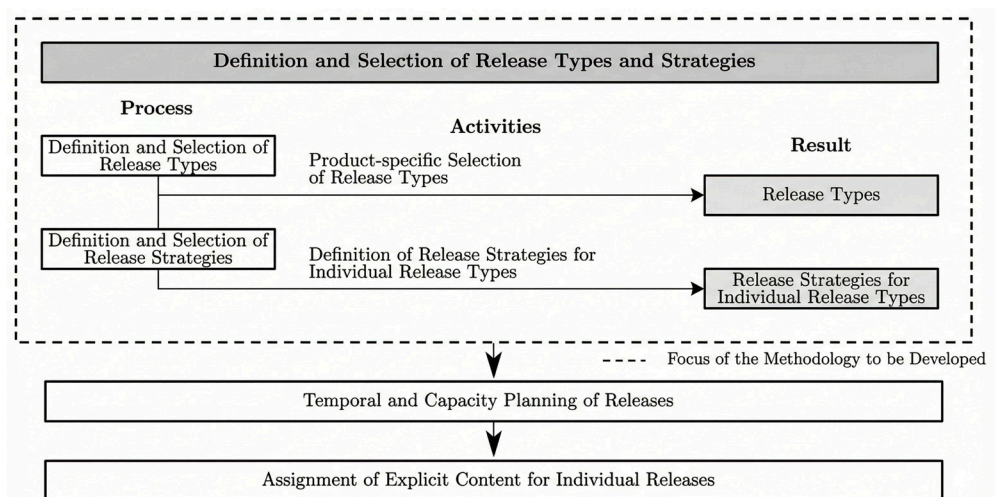


Figure 3. Reference process for defining release types and their strategies in the context of the methodical approach derived from Sahin et al. [67].

3.2. Definition and Selection of Release Types

Effective release planning requires detailed knowledge of product properties and dependencies, making the definition and selection of release types essential. After introducing release types from literature and practice, the methodology establishes a standardized in-

formation basis and an indicator-driven analysis approach that uses component complexity to support product-specific release-type selection.

3.2.1. Release Types in Literature and Practice

Release types must be differentiated because increasingly heterogeneous product structures complicate upgrade planning for products that integrate components from multiple domains and therefore require domain sensitive handling. A systematic overview is required because the basic distinction between Major Releases, Minor Releases, and Emergency Releases is no longer sufficient for current product and market complexity. Major Releases comprise substantial upgrades with high planning effort and resource consumption, Minor Releases comprise smaller and more flexible updates, and Emergency Releases comprise very small and urgent interventions triggered by unforeseen needs. Based on extensive analyses in the automotive, smartphone, and appliance industries, Şahin et al. extend this view by identifying release type differences at the levels of whole products, platforms, and modules. Based on extensive analyses in the automotive, smartphone, and appliance industries, Şahin et al. extend this view by identifying release type differences at the levels of whole products, platforms, and modules [67].

Sondereditions Releases provide differentiated variants with special design or feature packages that are often offered in limited numbers, which can stimulate demand near the end of the lifecycle, as illustrated by the Mercedes AMG E63S Final Edition [68]. Product maintenance Releases target quality, reliability, and cost-related optimizations and include software patches or security updates such as those for iOS or Windows [67]. Emergency Releases address unplanned corrective actions such as recalls triggered by defects or regulatory requirements, which is illustrated by the Porsche 911 GT3 seatbelt recall [69].

Figure 4 provides a qualitative comparison of scope, implementation effort, environmental volatility, and contribution to product value. Product generation and Product improvement Releases enable substantial value creation, whereas Sondereditions Releases can induce customer dissatisfaction when limited availability conflicts with expectations. Security-related and Emergency Releases primarily restore baseline product states rather than creating additional value. The subsequent methodology focuses on Product generation, Product improvement, and Product maintenance Releases because they exhibit the highest plannability, while the remaining types mainly reflect short term market impulses or marketing actions and are therefore documented but not central.

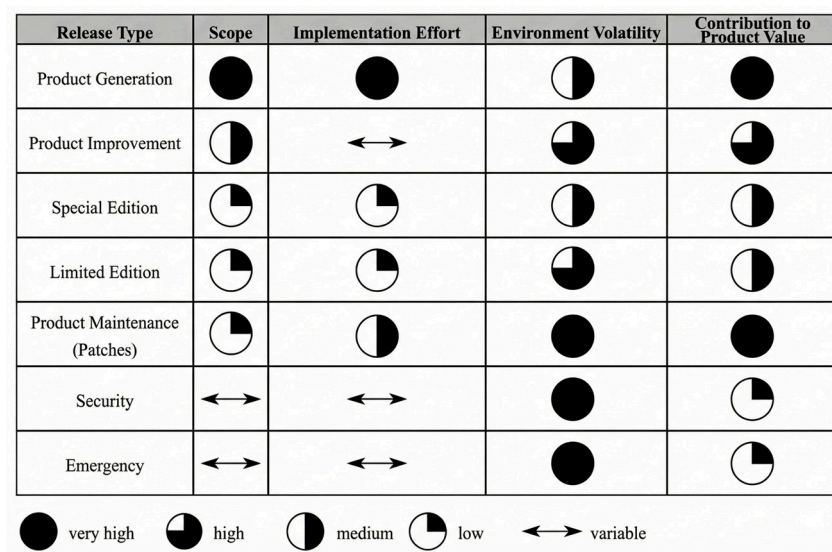


Figure 4. Criteria-based overview of release types in the literature.

3.2.2. Standardized Information Base as Input for the Methodology

Selecting product-specific release types requires a deep understanding of the product and its substance; a purely global view cannot reflect the diversity and potential of release planning. A solid foundation is therefore essential. Representing the product in its modular structure serves as the input for release-type selection, ensuring that product characteristics are consistently linked to release decisions. Product structures transparently depict components and their interdependencies and thus form the methodological reference point. Focusing on clearly defined modules with functional boundaries and interfaces provides an objective, consistent data basis for subsequent methodological steps, as shown in Figure 5.

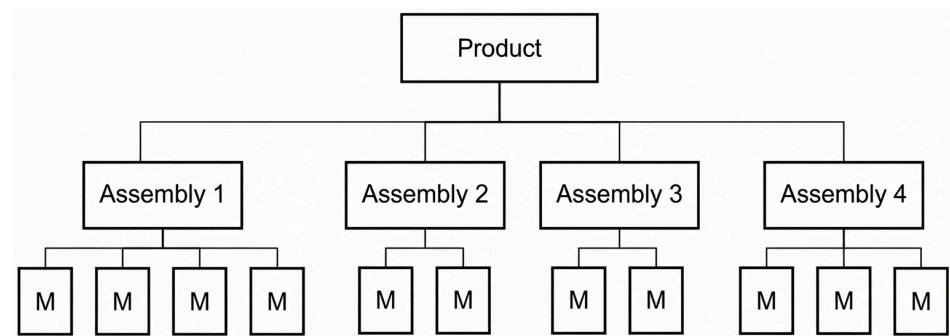


Figure 5. Exemplary product structure as input for the product-specific selection of release types (M: Module).

3.2.3. Criteria-Oriented Analysis of the Product Structure

Without detailed product insight, every release type would default to the whole product. Using the product structure enables a more granular selection by identifying critical modules whose characteristics open new release-type perspectives. This requires a criteria-based analysis of partial component complexity, grounded in four dimensions—technology, size, organization and environment [70]. The technology dimension reflects the number and maturity of technologies used in a product component, while size refers to the number of interacting components or domains within a module. Organization captures the number of involved people and departments, and environment describes market innovation speed, competitive dynamics and customer sensitivity. In this context, the criterion “Market/Competitive Dynamics” captures competitive intensity and the density of alternative offers as a structured indicator of external pressure to accelerate or re-prioritize upgrades. Although literature lists extensive complexity criteria [67,70,71], practical applicability requires a condensed set. These criteria are qualitatively evaluated by experts using an intentionally coarse, integer rating scale to avoid neutral responses. Table A1 summarizes criteria and scales. Using this catalog, experts assess components from product level downward across assemblies and modules, producing complexity profiles for each structural layer. This provides the data basis for selecting appropriate release types, as outlined in Figure 6.

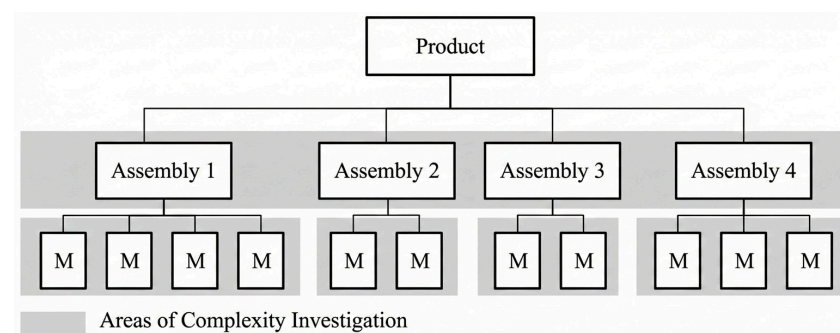


Figure 6. Assessment procedure for determining the complexity of product components (M: Module).

3.2.4. Selection of Release Types

After determining component complexities, these profiles are matched with suitable release types. A release data sheet enables this comparison by transforming complexity criteria into release-type-specific attributes, for example, mapping interacting domains or components to the criterion “scope.” It also lists the responsible department and relevant product-structure excerpt. Existing sheets for product-generation, product-improvement and product-maintenance releases support this process, illustrated in Figure 7.

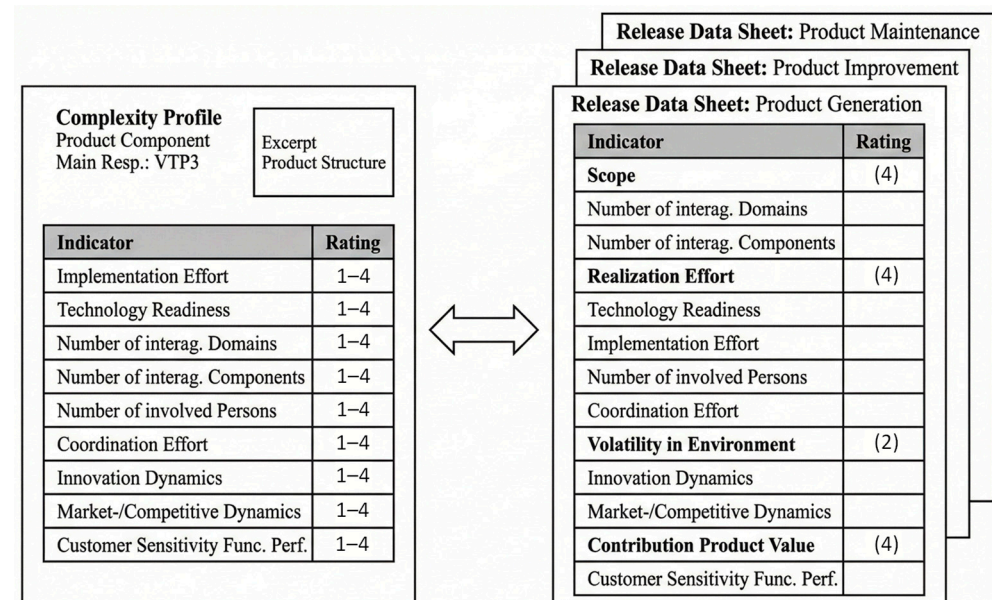


Figure 7. Procedure for comparing complexity profiles and release data sheets to determine release type recommendations.

The matching process is conducted systematically to ensure product-independent applicability. Starting at the highest product-structure level, each component is evaluated for complexity; the most complex element is marked and stored in a parallel “complexity memory” for later reference. Each component’s complexity profile is then compared with the three release-type data sheets to determine which release type it aligns with most closely. The transformed complexity values are matched against release-type criteria, producing a percentage-based congruence score. Components are then placed in a release-type pool, divided into the three categories and displaying their degree of alignment. A threshold of 70% indicates strong congruence and serves as guidance for final assignment of components to release types. This threshold represents an empirical value derived from practical experience; while it has proven effective in application, it acts as a heuristic that may warrant further examination in future research. The overall process is illustrated in Figure 8.

The procedure is repeated until reaching a product-structure level of low complexity, where components cannot be further subdivided and can be assessed as complete units. The steps generate a comprehensive overview of product elements, their complexities and the corresponding release-type criteria. The resulting release-type pool contains all evaluated components and supports final selection. Experts review the analysis and make decisions in coordination with stakeholders, guided by questions such as whether components can be bundled into one release type, whether specific domains require emphasis, or whether particular elements demand special handling. The chosen release types form release profiles, later expanded with strategy dimensions. This provides structured, product-based support for defining and selecting release types.

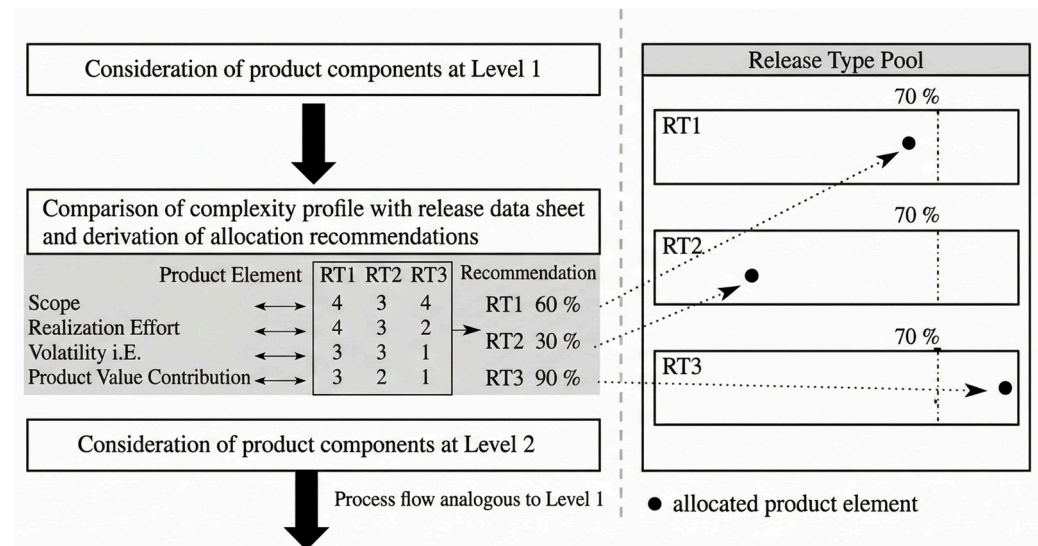


Figure 8. Transformation of the complexity of product components to derive recommendations for the release type pool.

3.3. Definition and Selection of Release Strategies

Successful product upgrades require not only product-specific release types but also systematic publication through well-defined release strategies, which guide and structure the planning process. To develop methodological support, existing strategy-definition approaches are introduced, which presents Şahin's core concept. This serves as the foundation for adapting and deriving release strategies within the methodology developed in this work.

3.3.1. Release Strategies in Literature and Practice

Practical observations show that products across industries may share similar release types, yet their publication strategies differ significantly. Automotive companies rely on long, predefined cycles, whereas household-appliance manufacturers react ad hoc with short-notice upgrades in response to changing customer needs [72]. A universal release strategy is therefore unsuitable; instead, strategies must be tailored to the specific characteristics of each release type. In a corporate context, strategy refers to coordinated actions aimed at achieving overarching goals and represents the outcome of formal planning processes [73]. Consequently, release strategies must be methodically derived. To ensure precision, their dimensions and possible specifications, outlined in Figure 9, must be clearly defined for the methodology.

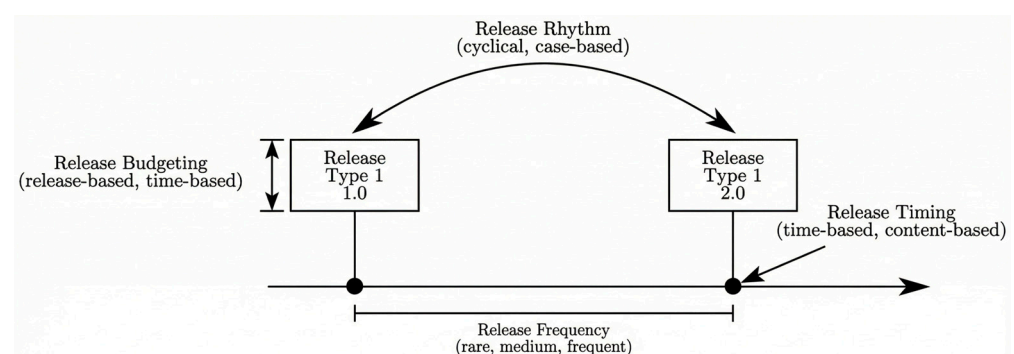


Figure 9. Design factors of release strategies and their dimensions along the release plan derived from Sahin et al. [67].

The literature provides clear interpretations of the dimensions defining release strategies. The planning horizon specifies the timeframe for a release type and must be set at the start of planning [6]. It may be fixed, requiring binding and unchangeable release schedules suitable for highly predictable release types, or variable, allowing temporary planning cycles that suit low-complexity releases requiring flexibility [67]. Release budgeting allocates limited capacities either per release or per period; extensive, complex releases benefit from release-specific budgeting due to improved transparency [1,67]. Release timing determines how release dates are set and must balance early market entry with technical maturity, a dilemma well known in software contexts. Timing can be time-based, following strict calendars, or feature-based, bundling interdependent features [58,67]. Release rhythm defines periodic (cyclic) or event-driven (case-based) placement of releases, with cyclic rhythms fitting products demanding continuous upgrades, and case-based rhythms fitting low-variability release types [67]. Release frequency reflects how often releases occur and depends on market and technology dynamics [18]; software releases typically show higher frequencies than multi-domain product upgrades [4]. Real-world examples illustrate these dimensions: Apple synchronizes hardware and iOS upgrades cyclically [72]; Microsoft uses structured generations combined with monthly patch cycles [72]; Vorwerk increases frequency through software-driven Thermomix upgrades [72]; German premium automakers follow long, predefined cycles [72]; Tesla adapts software-style rhythms, raising release frequency through over-the-air updates [72]. Different products and industries employ divergent release strategies, as shown in Figure 10, raising the unresolved question of which factors allow deriving suitable strategies for specific release types.

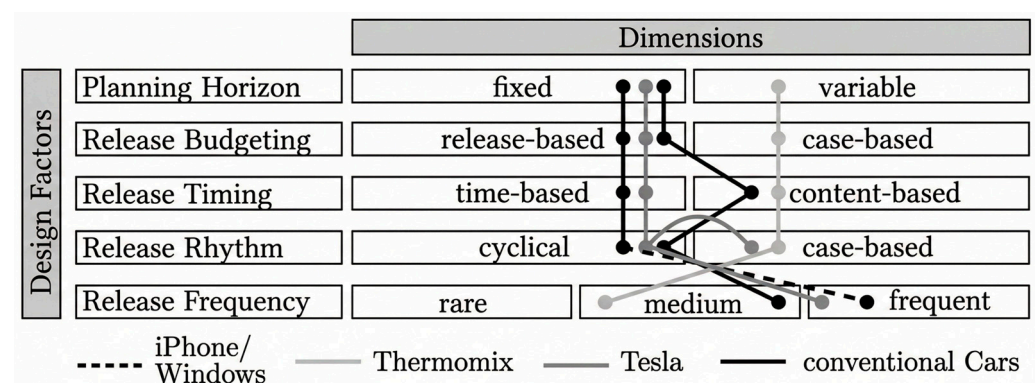


Figure 10. Different forms of release strategies for real products.

3.3.2. Methodical Support for the Derivation of Release Strategies

The wide range of possible release-strategy configurations highlights the difficulty of defining strategies that reflect the specific character of each release type. Methodological support is therefore needed to derive strategy recommendations based on clear orientation factors. Insights from project management show that increasing complexity requires iterative, smaller planning steps [74], a principle transferable to release planning. Accordingly, strategy selection must reflect the complexity of the release type. Schuh proposes a complexity classification based on diversity and volatility, distinguishing simple, complicated, relatively complex and highly complex systems [8]. By integrating this complexity theory with strategic planning, Şahin adapts this logic and defines four release strategies differing in change probability and implementation effort [67]. This theoretical linkage allows for the derivation of management strategies directly from the attributes of the product components. Low change and low effort favour flexible strategies; high change with low effort suggests resource-oriented strategies; low change with high effort requires static strategies; and high change combined with high effort demands highly systematised strategies. Figure 11 summarises this matrix-based classification.

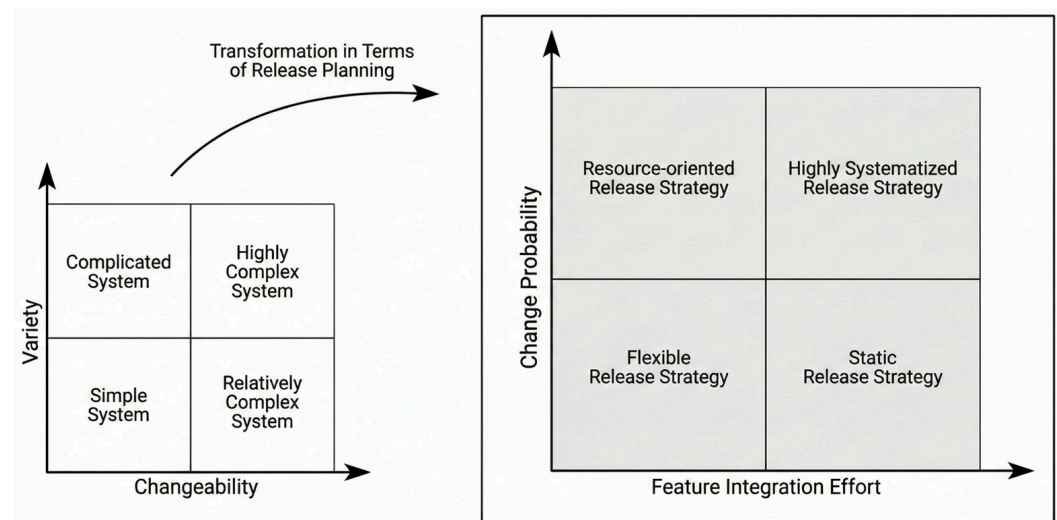


Figure 11. Transformation of Schuh's classification model for structuring release strategies derived from Sahin et al. [67].

The complexity of each release type is assessed using the indicators previously described, mapped to the two strategy-classification criteria. Their relationships are summarized in Table 1. This assessment provides initial guidance for selecting suitable strategies. The defined dimensions, characteristics and selection methods enable strategy choice for each release type, serving as input for later release-plan detailing and capacity allocation, though these subsequent steps fall outside the scope of this work.

Table 1. Assigning complexity criteria to the components of the release strategy.

Complexity Criterion	Release Strategy Component
Technology Maturity	Change Probability
Market/Competitive Dynamics	
Customer Sensitivity to Functional Fulfilment	
Innovation Dynamics	
Number of Interacting Domains	Feature Integration Effort
Number of Interacting Components	
Number of People Involved	
Implementation Effort	
Coordination Effort	

3.4. Methodology for Defining Release Types and Their Strategies

The preceding sections have successively developed a comprehensive method enabling the definition and selection of release types and their strategies. A formalized product-structure representation provides consistent input data, while a criteria-based complexity analysis allows matching product components with suitable release types and deriving explicit assignment recommendations. These complexity characteristics are then translated into strategy dimensions to determine appropriate release strategies. The result is a sequential, product-specific process that identifies both release types and corresponding strategies. Figure 12 summarizes all methodological steps.

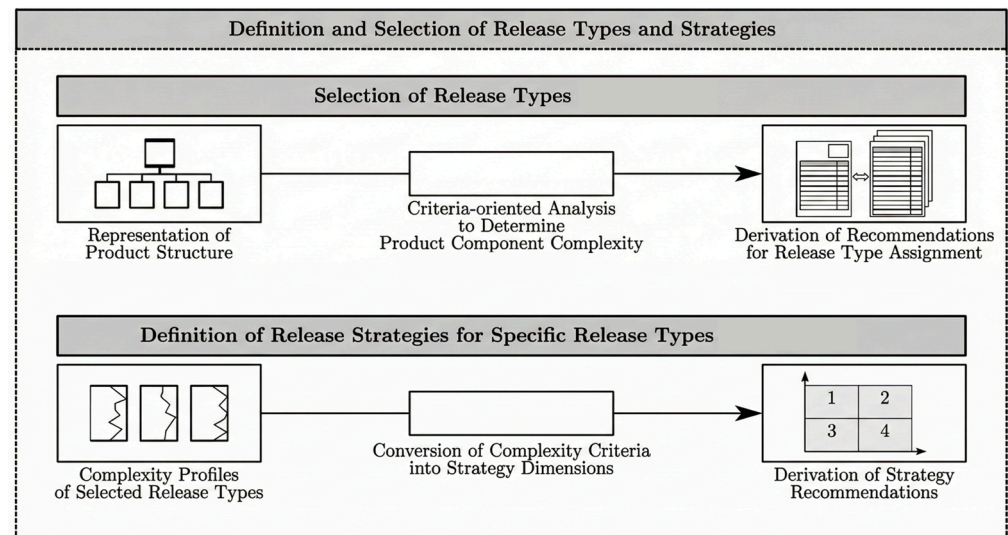


Figure 12. Overall overview of the methodology developed.

4. Validation and Evaluation of the Method

The developed methodology for defining and selecting release types and strategies requires validation of its functionality, usability and benefit. This section conducts a practical evaluation by defining relevant boundary conditions and applying the method to a modern electric vehicle. After outlining the upgrade problem, the method is tested for release-type and strategy selection, followed by presentation of the evaluation results.

4.1. Procedure for Evaluating the Methodology

Designing a seemingly functional methodological process is insufficient without proving that it delivers the intended benefit. Blessing et al. identify two essential aspects of method validation: application evaluation, which tests whether the method fits its intended context, and success evaluation, which examines its effectiveness [75]. Realistic assessment is crucial, requiring strong real-world relevance. However, full real-world application is difficult because the method depends on internal company data from multiple departments, some easily accessible and others requiring extensive effort. The strategic nature of the method also links it closely to product and corporate strategy, demanding long-term observation. Therefore, validation is performed using a modern electric vehicle as a realistic example, enabling demonstration of the method without excessive data collection and allowing full execution of release-type and strategy selection.

4.2. Application of the Methodology Using the Example of a Modern Electric Vehicle

The following section applies the developed methodology by first defining the scope of the chosen example. It then executes both main process steps while illustrating the Excel tool's operation, culminating in the selection of release types and derivation of suitable release strategies.

4.2.1. Initial Situation: E-nnovator GmbH and E-Rebel

The automotive industry is undergoing rapid transformation as conventional combustion vehicles give way to electric alternatives, accompanied by rising customer expectations. Cars increasingly function as experiential spaces rather than mere transport. In this environment, E-nnovator GmbH aims to position its E-Rebel as a modern electric vehicle and leverage release planning for continuous product development. Both corporate and product structures have been modularized, with 1000 employees organized into interdisciplinary departments aligned with vehicle components. The E-Rebel's product structure

follows the logic shown in Figure 13, providing a foundation for systematic upgrades and long-term competitiveness.

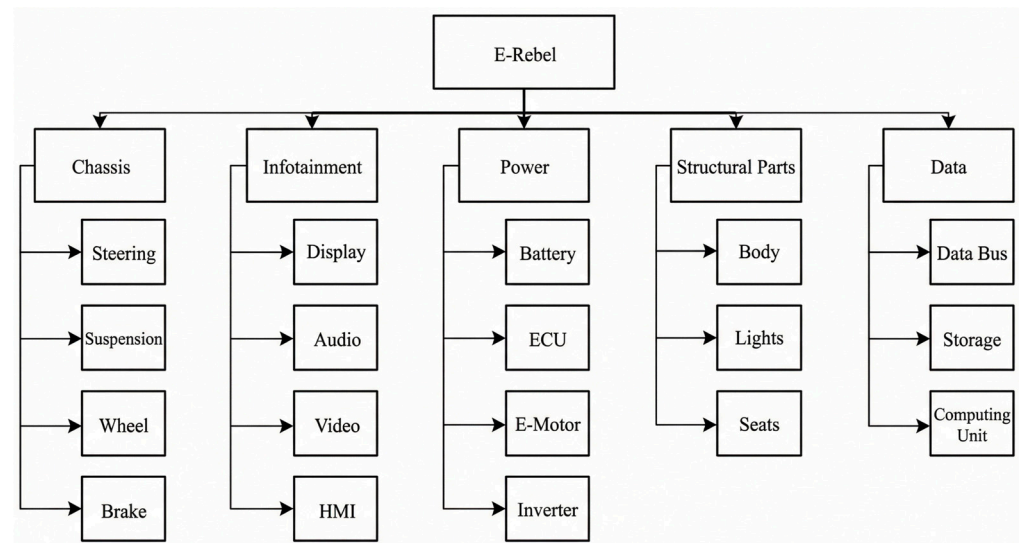


Figure 13. Product structure of the E-Rebel electric vehicle with assembly and module levels.

The E-Rebel is structurally divided into five main assemblies with underlying module levels. Its chassis ensures vehicle mobility, while the power unit provides electric drive. The infotainment assembly comprises interior components that enable information display and communication. Structural parts define the vehicle's exterior and interior shape, and the data-system assembly contains all elements for data processing and autonomous functions. Functional or feature-level descriptions are still under development. The E-Rebel competes in a highly dynamic market with numerous entrants and rapid innovation, particularly in battery technology, which demands significant development resources. Growing expectations regarding in-car entertainment further increase upgrade needs, and early market introduction has led to many customer complaints due to limited product maturity. To address these challenges, E-nnovator GmbH aims to establish a robust foundation for continuous improvement through the developed methodology, making the product-specific selection of release types and strategies a strategic priority.

4.2.2. Definition und Auswahl von Release-Typen E-Rebel

Classifying the E-Rebel into specific release types is essential for subsequent planning, so E-nnovator GmbH has agreed to adopt three global categories: generation, improvement and maintenance releases. Table 2 summarizes these types descriptively, with further specification applied later during the methodology's detailed implementation.

Table 2. Planned release types for E-Rebel from E-nnovator GmbH.

	Generation Release	Improvement Release	Maintenance Release
Release Description	Extensive renewal of the product or individual product components through the introduction of new technologies or expansion of existing concepts.	Introduction of innovations with high customer perception after the publication of Generation Releases through visual, functional, and performance-related upgrades of the product substance.	Provision of additional market impulses through the presentation of upgrades in the further life cycle with smaller scope but still related to quality and performance improvements.

The first step, defining process parameters, establishes the framework for applying the methodology. This includes setting the characteristics of release data sheets and weighting factors from the complexity assessment to guide smaller yet impactful lifecycle upgrades related to quality and performance.

Using a four-level scale, the three planned release types are assessed regarding scope, implementation effort, environmental volatility and contribution to product value, as shown in Figure 13. E-nnovator GmbH assigns these ratings through product-management experts. For the maintenance release, experts determine a low scope (1), moderate implementation effort (2), high environmental volatility (4) and a high value contribution (4). The same procedure is applied to generation and improvement releases to define their baseline characteristics. In parallel, the company evaluates the weightings of complexity indicators relative to the release-type classification criteria. For example, implementation effort is judged to depend 30% on technology maturity, 30% on implementation workload and 20% on personnel and coordination needs. All parameters are entered through the method's input interface, as shown in Figure 14.

1. Define Process Parameters			
Define Process Parameters	Generation	Improvement	Maintenance
Scope	4	2	1
Realization Effort	4	2	2
Volatility in Environment	2	3	4
Contribution to Product Value	4	3	4

Release Data Sheet		Weighting
Scope	1	
Number of interac. Domains		0.5
Number of interac. Components		0.5
Realization Effort	1	
Technology Maturity		0.3
Implementation Effort		0.3
Number of Involved Persons		0.2
Coordination Effort		0.2
Volatility in Environment	1	
Innovation Dynamics		0.5
Market/Competition Dynamics		0.5
Contribution to Product Value		
Customer Sensitivity to Function Fulfillment		1

Figure 14. Determination of process parameters within the scope of the Microsoft Excel application tool.

In Step 2, as shown in Table 3, determining component complexity, the complexity of all product elements is assessed. Since E-nnovator GmbH maintains detailed internal documentation, all teams understand the E-Rebel's product structure and support the evaluation. Using the criteria catalog (Table A1), experts rate assemblies and modules. For example, the chassis receives a complexity score of 3 due to interacting technologies and high implementation effort. Technology maturity is rated 2, as most technologies are established but suspension systems require further development. Strong domain interaction and extensive interdependencies yield maximum scores, as does the high share of employees working in chassis development. Limited market innovation results in ratings of 2, while customer sensitivity is rated 3. Conducting this evaluation across all assemblies and modules provides a complete complexity profile of the E-Rebel.

In Step 3, matching component complexity with release-type specifics, the previously generated complexity profiles are compared with the characteristics of each release type. Using the process parameters defined in Step 1, the methodology calculates a percentage-based fit between each component's complexity and the requirements of the three release types. For the chassis example, the method yields 84.2% alignment with the generation release, 65.8% with the improvement release and 40.8% with the maintenance release, as shown in Table 4. This comparison is performed for all E-Rebel components, with the tool-assisted workflow illustrated in Appendix A.

In Step 4, reviewing the release-type pool and making initial assignments, the calculated component–release-type matches are consolidated into an overall overview. The pool, divided into the three planned release types, now contains all E-Rebel components with

their respective match percentages. This enables E-nnovator GmbH to conduct an expert discussion and make preliminary assignment decisions. Figure 15 shows the resulting release-type pool.

Table 3. Transfer of the chassis complexity profile as part of the application of the Microsoft Excel tool.

Chassis	
Implementation Effort	3
Technology Maturity	2
Number of Interacting Domains	4
Number of Interacting Components	4
Number of Involved Persons	4
Coordination Effort	4
Innovation Dynamics	2
Market/Competitive Dynamics	2
Customer Sensitivity to Function Fulfillment	3
Total	28

Table 4. Interim result of the Microsoft Excel tool for comparing complexity profile and release type data sheets.

Chassis		Generation	Improvement	Maintenance
Implementation Effort	3	0	2	3
Technology Maturity	2	0.9	1.1	1.1
Number of Interacting Domains	4	0	1	2
Number of Interacting Components	4	1	0	1
Number of Involved Persons	4	84.2%	65.8%	40.8%
Coordination Effort	4			
Innovation Dynamics	2			
Market/Competitive Dynamics	2			
Customer Sensitivity to Function Fulfillment	3			
Total	28			

Generation		
Module	Agreement	
Bodywork	85,8%	
Steering	85,8%	
Chassis	84,2%	
Structural parts	83,3%	
Power	77,5%	
Battery module	75,8%	
Infotainment	75,8%	
Brake	75,0%	
Audio	71,7%	
Data	72,5%	
Data storage	61,7%	
Central Processing Unit	61,7%	
Data Bus	61,7%	
E-Machine	60,0%	
Display	56,7%	
Suspension	56,7%	
Wheel	55,8%	
Lights	55,0%	
Seats	54,2%	
ECU	53,3%	
Video	53,3%	
HMI	52,5%	
Inverter	52,5%	

Improvement		
Module	Agreement	
ECU	96,7%	
Lights	88,3%	
Display	85,0%	
Suspension	85,0%	
Inverter	80,8%	
HMI	80,8%	
Seats	79,2%	
Audio	78,3%	
Data	77,5%	
Wheel	72,5%	
Data storage	71,7%	
Central Processing Unit	71,7%	
Data Bus	71,7%	
Structural parts	66,7%	
Brake	66,7%	
Chassis	65,8%	
Bodywork	64,2%	
Steering	64,2%	
Video	63,3%	
Battery module	57,5%	
Infotainment	57,5%	
E-Machine	56,7%	
Power	55,8%	

Maintenance		
Module	Agreement	
HMI	89,2%	
Inverter	72,5%	
Wheel	72,5%	
ECU	71,7%	
Audio	70,0%	
Display	68,3%	
Infotainment	65,8%	
Battery module	65,8%	
Power	64,2%	
Lights	63,3%	
Steering	55,8%	
Suspension	60,0%	
Brake	58,3%	
Structural parts	58,3%	
Bodywork	55,8%	
Seats	54,2%	
Data	52,5%	
Data storage	46,7%	
Central Processing Unit	46,7%	
Data Bus	46,7%	
Chassis	40,8%	
Video	38,3%	
E-Machine	31,7%	

Figure 15. Release type pool of the Microsoft Excel tool for the E-Rebel application example.

Within the release-type pool, components are sorted by descending match percentage. E-nnovator GmbH applies a 70% threshold (green in Figure 14), above which an assignment is considered sufficiently accurate and forwarded to the final expert decision round, ensuring stakeholder knowledge is incorporated. Using the pool, product elements are now allocated to release types. While the whole product could be treated as a single release unit, the goal is a more granular, product-specific differentiation. For generation releases, all major assemblies—chassis, structural parts, power, infotainment and data—exceed the 70% threshold, indicating that a full product-generation upgrade should include all of them. Several modules (body, steering, battery, brakes) also fall into this category, confirming their high complexity and need for inclusion exclusively in generation releases. In the improvement-release pool, the data assembly appears in both the generation and improvement categories; due to rapid technological change, E-nnovator selects it for a software-based improvement release to introduce customer-oriented features during the lifecycle. Multiple hardware modules (lights, display, suspension, inverter, seats, wheel and audio) also meet the threshold and are planned as hardware improvement releases. Software-dominant modules such as ECU and HMI form a dedicated software improvement release enabling upgrades like new driving modes or gesture-based infotainment control. Maintenance releases show fewer high matches, but four modules qualify: HMI, ECU and the central computing unit are grouped into a software maintenance release, while the wheel module is planned for hardware maintenance releases to introduce quick market-stimulating design updates. Table 5 summarizes all assignments.

Table 5. Initial release profiles after initial assignment of product components to release types.

	Product Generation Release	Improvement Release Data	Improvement Release Software
Release Description	Extensive upgrade of the E-Rebel focusing on all vehicle assemblies. This includes significant changes in appearance and performance to set new market impulses.	Rapid developments in the Data assembly area require focused assembly area require focused software-based upgrades for the entire assembly. This implies extensive updates for the E-Rebel autopilot and the expansion of existing data storage systems.	Due to expected high dynamics in the area of engine control (ECU) and the infotainment operating concept (HMI), new features on a purely software basis will be placed on the market for upgrading.
Release Unit	Complete product including all assemblies	Data Assembly	ECU, HMI
Release Domain	cross-domain	Software	Software
Release Perception	very high	high	medium
	Improvement Release Hardware	Maintenance Release Software	Maintenance Release Hardware
Release Description	Realization of customer-oriented upgrades of multiple hardware components of the E-Rebel to create a lasting influence on long-term product value.	Particularly high dynamics in the area of modules that provide software-based features require the publication of maintenance releases. Small functional improvements for the affected units can be bundled in them.	Short-term impulses in the E-Rebel's life cycle can be realized through less extensive upgrades of the product module 'Wheel'. Here, the introduction of a new design or a new color scheme is conceivable.
Release Unit	Lights, Display, suspension, Inverter, seats, Wheel, Audio	ECU, HMI, Central Processing Unit	Wheel
Release Domain	Hardware	Software	Hardware
Release Perception	medium	low	medium to high

This completes the definition and selection of release types for E-Rebel. In the next step, the release types defined will serve as the starting point for defining associated release strategies.

4.2.3. Derivation of Individual Release Strategies E-Rebel

To publish the selected release types successfully, E-nnovator GmbH must derive strategies that determine how each release unit enters the market. Following the clustering approach described previously, strategies are defined based on change probability and feature-integration effort. The Excel tool supports experts by calculating percentage values for both indicators, guiding placement within the strategy matrix. For the product-generation release, which includes all assemblies, high values emerge: 80% change probability and 84% integration effort. In contrast, the hardware maintenance release shows much lower values, 62.5% and 40%, highlighting the need for tailored strategies. Table 6 summarizes the results.

Table 6. Analysis results for the indicators of the strategy definition within the Microsoft Excel tool.

Release Type	Change Probability	Effort for Feature Integration
Product Generation Release	80.00%	84.00%
Improvement Release Data	75.00%	70.00%
Improvement Release Software	87.50%	50.00%
Improvement Release Hardware	61.61%	54.29%
Maintenance Release Software	91.67%	55.00%
Maintenance Release Hardware	62.50%	40.00%

The release types show varying strengths across both strategy criteria. For software and hardware improvement releases and the software maintenance release, feature-integration effort lies near the 50% mark, making quadrant assignment less distinct and requiring careful stakeholder consideration. Mapping the percentage values in Figure 16 enables final placement within the defined strategy quadrants.

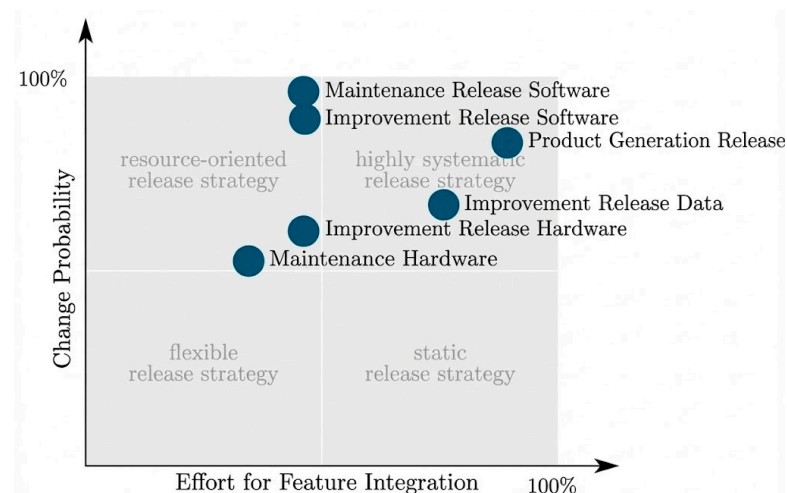


Figure 16. Individual release strategies of the E-Rebel.

Given the high change probability and equally high feature-integration effort, the product-generation release requires a highly systematized strategy. Release dates and capacity allocations must be planned far in advance to avoid excessive resource use, and the high volatility necessitates cyclic introduction. This level of systematization also implies a fixed planning horizon with limited flexibility. The Data improvement release likewise demands a highly systematized strategy, though its lower effort allows shorter upgrade cycles. Improvement and maintenance releases follow a resource-oriented strategy; al-

though calculations place them in the highly systematized quadrant, E-nnovator GmbH prioritizes flexibility and assigns resources individually per release, enabling higher frequencies due to lower integration effort. Software-based releases offer additional agility. With the completion of release profiles, strategy derivation is finalized, though continuous reassessment remains necessary. Table A2 presents the resulting profiles, which now serve as input for subsequent planning steps beyond this work's scope. A deeper analysis of these results in comparison to existing practices highlights the method's specific impact. While traditional automotive strategies often force a synchronization of all domains into rigid lifecycles (e.g., facelifts), the strategies derived here explicitly decouple the 'Data' and 'Software' components from the 'Chassis'. By assigning a 'resource-oriented' and 'highly systematized' strategy with higher frequency to the former, the E-Rebel avoids the common pitfall of delaying software innovations to match hardware timelines. This demonstrates the method's capability to generate product-specific strategies that outperform standardized, one-size-fits-all approaches.

4.3. Evaluation of the Method

The developed methodology must be evaluated against the requirements defined in Section 3.1 to verify its effectiveness. It fully supports an extensive planning process, guiding users from detailed product-structure preparation to the derivation of release strategies. The method strongly incorporates product complexity by comparing component-level complexity profiles with release-type characteristics. Modular product structures are explicitly addressed, and their formal representation serves as standardized input for the procedure. The three release types—generation, improvement and maintenance—are clearly defined and linked to product specifics. The method also enables the derivation of alternative release strategies based on criteria such as change probability and integration effort. Product specificity is ensured through component-focused complexity assessment. After executing the method, a structured release plan can be outlined, supported by release profiles and strategy recommendations covering planning horizon, budgeting, timing, rhythm and frequency. Market dynamics are incorporated through complexity criteria that reflect innovation levels and competitive movement. Overall, all requirements for defining and selecting release types and strategies are fulfilled, demonstrating that the methodology provides an effective, practical instrument for efficient, product-specific release planning and long-term value enhancement.

5. Discussion

The primary objective of this work was to develop a methodology for defining and selecting release types and strategies, addressing the specific challenges of software-intensive, modular products. The application of the method to the E-Rebel use case demonstrates that linking product substance to release-planning implications effectively bridges the gap between structural modeling and strategic planning identified in the state of the art. Market competition also affects product upgrading because it influences both the urgency and the positioning function of updates. High competitive intensity can increase customer expectations, reduce tolerance for functional gaps, and elevate the strategic relevance of timely releases as signals of innovativeness, which directly affects release frequency and timing decisions [4–6]. Although the proposed methodology does not perform an explicit competitor benchmarking step, competitive pressure is incorporated through the environment-related complexity criteria, particularly "Market/Competitive Dynamics," and thus affects release type and strategy recommendations via expert assessment. For applications in highly contested markets, a lightweight competitor scan can be used to calibrate ratings and thresholds without changing the core procedure [5,6].

While previous approaches, such as Maurer et al., focus primarily on cycle-oriented timing for hardware, and Kühn offers a framework for mechatronic systems, our results indicate that a purely cycle-based view is insufficient for modern hybrid products [4,11,37,55]. The E-Rebel case study confirmed that software components (e.g., HMI, ECU) require fundamentally different release strategies (resource-oriented or flexible) compared to hardware assemblies (e.g., Chassis), which demanded highly systematized strategies. This aligns with Broy's observation regarding the distinct temporal patterns of automotive software and Porter and Heppelmann's description of the shift toward intelligent systems [22,24,47]. By operationalizing Sahin's strategic framework, this methodology provides the missing link: a quantifiable connection between Schuh's concept of product complexity and the selection of specific release types [8,51,54,59,67,72,76].

The evaluation relies on complexity as the central decision-making indicator, condensed into a reduced set of criteria to ensure usability. While Erixon utilizes similar complexity drivers for modularization in Modular Function Deployment (MFD), our method extends this logic to the lifecycle phase [40]. However, this condensation limits objectivity. In practice, a broader set of influences must be considered, such as cost structures and technological feasibility, factors highlighted by Ehrlenspiel et al. as critical to development but excluded here to maintain methodological focus [35].

Furthermore, the input values in the case study relied on expert estimations. While this is consistent with the expert-driven weighting approaches found in Göpfert's METUS procedure, real industrial data would strengthen validity [33,43]. The resulting six release profiles represent strategic proposals rather than fixed plans and must be continuously adapted to market dynamics, echoing Nippa and Labriola's emphasis on the need for flexible Time-to-Market management [5]. Additionally, the application excluded competitor strategies, although Hepperle notes these strongly influence release timing [6]. Finally, only a single product, the E-Rebel, was examined; future research should address Krause and Gebhardt's focus on modular product families, as structural linkages across a portfolio could significantly alter the optimal selection of release types and strategies [39].

6. Conclusions

Today's market environment is more dynamic than ever. Rapid technological progress and a growing number of competitors intensify customer expectations and increase the pressure on companies to offer product structures that maximize customer value while accommodating volatile conditions. Sustainable success increasingly depends on products that continuously adapt to evolving market needs. Systematic planning of product upgrades, realized through structured release planning, offers a promising approach, enabling targeted steering of product value through defined upgrade units (release types). Because effectiveness hinges on product-specific release types and their associated publication strategies, this work aimed to develop a methodology that supports their systematic definition and selection, with a strong focus on practical applicability.

The methodology is founded on understanding product structures at a modular level and assessing component-specific complexity to determine suitable release types. It further provides strategic recommendations for the publication of these upgrade units, summarized in dedicated release profiles. Its practical feasibility is demonstrated through application to a modern electric vehicle, where product-specific release types and strategy proposals were derived based on structural and contextual factors. A subsequent evaluation confirms that the developed methodology fulfils all intended requirements and offers substantial support for systematic, value-oriented release planning. Consequently, this research enriches the body of knowledge in two distinct dimensions. Theoretically, it bridges the gap between modular product architecture and release management by demonstrating how component

complexity serves as a valid determinant for differentiating release types in software-intensive systems. Practically, it operationalizes this theoretical link into an applicable procedural framework, enabling companies to move beyond intuitive planning toward a systematic, data-driven definition of release strategies.

7. Outlook

The methodology developed in this work contributes to the systematic planning of product upgrades and enables fundamental recommendations for product-specific release organization, while also highlighting the complexity of release planning itself. Future research should particularly focus on the symbiotic relationship between product architecture and release planning to explore the potential of designing products fully aligned with release processes. Considering ad-hoc changes is equally important, as real-world factors, such as user feedback, regulatory shifts, or cost pressures, often require deviations from predefined plans. Identifying highly upgrade-relevant subsystems could further support more granular release planning. The growing share of software in products increases the importance of software-based upgrades and enables new business models such as “features on demand,” which create value for both customers and companies. As systematic release planning becomes increasingly critical across industries, the methodology proposed here provides essential impulses for structuring and operationalizing release management.

Author Contributions: Conceptualization, U.V.K. and A.S.; methodology, U.V.K., A.S. and M.A.; modeling, U.V.K. and A.S.; validation, A.S. and T.V.; formal analysis, U.V.K.; investigation, M.; writing—original draft preparation, A.S.; writing—review and editing, U.V.K. and A.S.; visualization, M.A.; supervision, T.V.; project administration, U.V.K.; funding acquisition, T.V. All authors have read and agreed to the published version of the manuscript.

Funding: The publication of this paper has been funded by the TU Braunschweig Publication Fund.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: This work was conducted in the context of the projects “ZL MOB” (Zukunftslabor Mobilität), which was funded by the “Zukunft. Niedersachsen” initiative of the Lower Saxon Ministry for Science and Culture and the Volkswagen Foundation and “ReTraSON” (Regionales Transformationsnetzwerk Südostniedersachsen), which is funded by the “Bundesministerium für Wirtschaft und Klimaschutz” (Federal Ministry for Economic Affairs and Climate Action). During the preparation of this manuscript/study, the author(s) used Google Gemini 3.1 for the purposes of translating text and adjusting text style. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Abbreviation	Definition
ECU	Engine Control (Unit)
FODA	Feature-Oriented Domain Analysis
HMI	Human Machine Interface (Operating concept of the infotainment)
LD	Linear dichroism
MDPI	Multidisciplinary Digital Publishing Institute
MFD	Modular Function Deployment
TLA	Three letter acronym
TTM	Time-to-Market

Appendix A

Table A1. Criteria catalog for determining the complexity of product components.

Criterion	Score	Criterion Characteristic
Implementation Effort	4	A multitude of different technologies interact with each other, little experience in the area
	3	A multitude of different technologies interact with each other, much experience in the area
	2	One technology with know-how, possibly also software possibilities
	1	Upgrade is realizable through software and requires therefore little resource effort
Technology Maturity	4	Use predominantly new technologies with little experience
	3	Use predominantly new technologies with extensive experience
	2	Use of mature technologies dominate
	1	Almost exclusive use of mature technologies
Number of Interacting Domains	4	Interaction of four domains
	3	Interaction of three domains
	2	Interaction of two domains
	1	Observation space only within one homogeneous domain
Number of Interacting Components	4	Multitude of interactions with superordinate product parts
	3	Two further interactions with additional superordinate product part
	2	One further interaction with additional superordinate product part
	1	No further interaction with additional superordinate product part
Coordination Effort	4	With several main departments/across departments
	3	With one further main department
	2	Within the own department
	1	Only within the own team
Number of Involved Persons	4	Over 25% of the persons involved in the product
	3	Less than or equal to 20% of the persons involved in the product
	2	Less than or equal to 15% of the persons involved in the product
	1	Less than or equal to 10% of the persons involved in the product
Innovation Dynamics	4	Rapid development on the market, very short innovation cycles, predominantly disruptive innovations
	3	Rapid development on the market, very short innovation cycles, incremental innovations
	2	Low innovation pressure
	1	Innovations not to be expected
Market /Competitive Dynamics	4	Volatile competitive situation, very many competitive offers available
	3	Manageable number of competitors with few competitive offers
	2	Important marketing function, an important player on the target market
	1	Complete marketing function, leading on the target market
Customer Sensitivity for Function Fulfillment	4	Customer perceives change in the function during use immediately true
	3	Customer perceives change in the function in the course of use true
	2	Customer perceives change in the function with great time delay true
	1	Customer perceives change in the function not true

Table A2. Results of the methodology for selecting and defining release types and their strategies using the example of E-nnovator GmbH.

	Product Generation Release	Improvement Release Data	Improvement Release Software
Release Description	Extensive upgrade of the E-Rebel with focus on all component groups of the vehicle. Particular emphasis is placed on changes in appearance and performance to create new market impulses.	Rapid developments in the data component group environment require focused upgrades on a software basis for the entire component group. This implies, for example, extensive updates for the E-Rebel autopilot as well as the expansion of existing data storage systems.	Due to expected high dynamics in the area of an engine control (ECU) as well as the operating concept of the infotainment (HMI), new features on a purely software basis are placed on the market for upgrading.
Release Unit	Complete product including all component groups	Data component group	ECU, HMI
Release Domain	cross-domain	Software	Software
Release Perception	very high	high	medium
Planning Horizon	fixed	fixed	variable
Release Budgeting	release-based	release-based	time-based
Release Timing	time-based	time-based	content-based
Release Rhythm	cyclical	cyclical	case-based
Release Frequency	moderate	high	moderate
	Hardware Improvement Release	Software Maintenance Release	Hardware Maintenance Release
Release Description	Realization of customer-oriented upgrades of several hardware components of the E-Rebel to generate a sustainable influence on the long-term product value.	Particularly high dynamics in the area of modules that provide software-based features require the publication of maintenance releases. Small functional improvements for the affected units can be bundled in these.	Short-term impulses in the lifecycle of the E-Rebel can be realized through a less extensive upgrade of the “Wheel” product module. Here, the introduction of a new design or a new color scheme is conceivable.
Release Unit	Lights, Display, Suspension, Inverter, Seats, Wheel, Audio	ECU, HMI, Central Processing Unit	Wheel
Release Domain	Hardware	Software	Hardware
Release Perception	medium	low	medium to high
Planning Horizon	variable	variable	variable
Release Budgeting	time-based	time-based	time-based
Release Timing	content-based	content-based	content-based
Release Rhythm	case-based	case-based	case-based
Release Frequency	moderate	moderate	moderate

References

1. Ruhe, G. *Product Release Planning: Methods, Tools, and Applications*; CRC Press: Boca Raton, FL, USA, 2010; ISBN 978-1-4200-0411-3.
2. Gausemeier, J.; Rammig, F.J.; Schäfer, W. *Design Methodology for Intelligent Technical Systems*; Springer: Berlin/Heidelberg, Germany, 2014; ISBN 978-3-642-45434-9.
3. Carlshamre, P. Release Planning in Market-Driven Software Product Development: Provoking an Understanding. *Requir. Eng.* **2002**, *7*, 139–151. [[CrossRef](#)]
4. Kühn, A.T. *Systematik zur Release-Planung Intelligenter Technischer Systeme*; Paderborn University: Paderborn, Germany, 2016.

5. Nippa, M.; Labriola, F. Der Roadmapping-Ansatz als integrative Planungsmethode im Rahmen eines situationsgerechten Time-to-Market Management. In *Technologie-Roadmapping: Zukunftsstrategien für Technologieunternehmen*; Möhrle, M.G., Ed.; Springer: Dordrecht, The Netherlands, 2007; pp. 297–324, ISBN 9783540747543.
6. Hepperle, C. *Planung Lebenszyklusgerechter Leistungsbündel*; Technische Universität München: München, Germany, 2013.
7. Albers, S.; Herrmann, A. (Eds.) *Handbuch Produktmanagement: Strategieentwicklung, Produktplanung, Organisation, Kontrolle*, 2nd revised and enlarged ed.; Springer Gabler: Wiesbaden, Germany, 2002; ISBN 9783663057536.
8. Schuh, G.; Riesener, M. *Produktkomplexität Managen: Strategien, Methoden, Tools*, 3rd revised and enlarged ed.; Hanser: München, Germany, 2017; ISBN 9783446452251.
9. Fricke, E.; Schulz, A.P. Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout Their Entire Lifecycle. *Syst. Eng.* **2005**, *8*, 342–359. [\[CrossRef\]](#)
10. Schubert, P. *Entscheidungsunterstützte Methodik zur Produktkonzeptauswahl: Grundlagen, Systematik und Exemplarische Anwendung*; Karlsruher Institut für Technologie: Karlsruhe, Germany, 2015.
11. Maurer, M.; Lindemann, U. (Eds.) *Zyklusorientiertes Modul- und Plattformdenken—Ein Leitfaden für Praktiker*; Eigenverlag: Berlin, Germany, 2014; ISBN 978-3-00-048286-1.
12. Bergmann, O. *Dynamik im IT-Markt: Unternehmenslebenszyklen im Halbleitersektor*; Universität Freiburg: Wiesbaden, Germany, 2000.
13. Wagner, R. *Projektmanagement in der Automobilindustrie*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2015; ISBN 978-3-658-08812-5.
14. Wohlin, C.; Aurum, A. Criteria for Selecting Software Requirements to Create Product Value: An Industrial Empirical Study. In *Value-Based Software Engineering*; Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., Grünbacher, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 179–200, ISBN 978-3-540-25993-0.
15. Aumayr, K.J. *Erfolgreiches Produktmanagement*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2019; ISBN 978-3-658-25365-3.
16. Esch, F.-R. *Strategie und Technik des Automobilmarketing*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2013; ISBN 978-3-8349-3391-1.
17. Herrmann, A.; Huber, F. *Produktmanagement*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2013; ISBN 978-3-658-00003-5.
18. Aleksic, S. *Nachhaltige Weiterentwicklung von Modulen Produktarchitekturen durch Release-Management*; RWTH Aachen: Aachen, Germany, 2016.
19. Lindemann, U. *Methodische Entwicklung Technischer Produkte: Methoden Flexibel und Situationsgerecht Anwenden*; 2nd bearb. Aufl.; Springer: Berlin/Heidelberg, Germany, 2007; ISBN 9783540374350.
20. Brockhoff, K. *Produktpolitik*; Fischer: Stuttgart, Germany, 1981; ISBN 3-437-40097-5.
21. Dudic, D. *Modell für die Fabrik Life Cycle-Orientierte Produktplanung und -Entwicklung*; Universität Stuttgart: Heimsheim, Germany, 2010.
22. Geisberger, E.; Broy, M. *agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems*; Springer: Berlin/Heidelberg, Germany, 2012; ISBN 978-3-642-29098-5.
23. Isermann, R. *Mechatronische Systeme: Grundlagen*; 2nd vollständig neu bearb. Aufl.; Springer: Berlin/Heidelberg, Germany, 2008; ISBN 978-3-540-32336-5.
24. Porter, M.E.; Heppelmann, J.E. How Smart, Connected Products Are Transforming Competition. *Harv. Bus. Rev.* **2014**, *92*, 64–88.
25. Sax, E.; Reussner, R.; Guissouma, H.; Klare, H. A Survey on the State and Future of Automotive Software Release and Configuration Management. *Karlsru. Rep. Inform.* **2017**, *11*. [\[CrossRef\]](#)
26. Feldhusen, J.; Grote, K.-H. (Eds.) *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung Erfolgreicher Produktentwicklung*; 8th vollständig überarb. Aufl.; Springer: Berlin/Heidelberg, Germany, 2013; ISBN 9783642295683.
27. Großklaus, R.H.G. *Von der Produktidee zum Markterfolg*; Gabler Verlag: Wiesbaden, Germany, 2014; ISBN 978-3-8349-4593-8.
28. Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.-H. *Konstruktionslehre: Grundlagen Erfolgreicher Produktentwicklung; Methoden und Anwendung*, 7th ed.; Springer: Berlin/Heidelberg, Germany, 2007; ISBN 9783540340607.
29. Schäuffele, J.; Zurawka, T. *Automotive Software Engineering*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2016; ISBN 978-3-658-11814-3.
30. Krüger, B.; Stein, A.; Gründker, L.; Vietor, T. Analyzing SME Digitalization Requirements Through a Technology Radar Framework in Southeast Lower Saxony. *Digital* **2025**, *5*, 60. [\[CrossRef\]](#)
31. Colares, F.; Souza, J.; Carmo, R.; Pádua, C.; Mateus, G.R. A New Approach to the Software Release Planning. In *Proceedings of the 2009 XXIII Brazilian Symposium on Software Engineering (SBES)*, Fortaleza, Ceara, Brazil, 5–9 October 2009; pp. 207–215, ISBN 978-1-4244-5024-4.
32. Schönsleben, P. *Integrales Logistikmanagement: Operations und Supply Chain Management Innerhalb des Unternehmens und Unternehmensübergreifend*; 6th bearb. und erw. Aufl.; Springer: Berlin/Heidelberg, Germany, 1998; ISBN 978-3-642-20381-7.
33. Göpfert, J. *Modulare Produktentwicklung: Zur Gemeinsamen Gestaltung von Technik und Organisation*; Deutscher Universitätsverlag: Wiesbaden, Germany, 1998; ISBN 978-3-8244-6827-0.
34. Ulrich, K. The Role of Product Architecture in the Manufacturing Firm. *Res. Policy* **1995**, *24*, 419–440. [\[CrossRef\]](#)

35. Ehrlenspiel, K.; Kiewert, A.; Lindemann, U. *Kostengünstig Entwickeln und Konstruieren: Kostenmanagement bei der Integrierten Produktentwicklung*; Springer: Berlin/Heidelberg, Germany, 2007; ISBN 978-3-540-74222-7.
36. Inkermann, D.; Hanna, M.; Richter, T.; Wortmann, N.; Vietor, T.; Krause, D. Die Produktarchitektur als zentrales Konzept in der Produktentwicklung. In *DFX 2019: Proceedings of the 30th Symposium Design for X, Jesteburg, Germany, 18–19 September 2019*; The Design Society: Glasgow, UK, 2019.
37. Maurer, M.; Bauer, W.; Elezi, F.; Chucholowski, N. Methodik zur Erstellung zyklengerechter Modul- und Plattformstrategien. In *Innovationsprozesse Zyklensorientiert Managen: Verzahnte Entwicklung von Produkt-Service-Systemen*; Vogel-Heuser, B., Ed.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 139–154, ISBN 9783662449318.
38. Belener, P.M. *Technisches Änderungsmanagement Modularer Produkte und Prozesse*; Ruhr-Universität Bochum: Aachen, Germany, 2008.
39. Krause, D.; Gebhardt, N. *Methodische Entwicklung Modularer Produktfamilien*; Springer: Berlin/Heidelberg, Germany, 2018; ISBN 978-3-662-53039-9.
40. Erixon, G. *Modular Function Deployment: A Method for Product Modularisation*; Kungliga Tekniska Högskolan: Stockholm, Sweden, 1998.
41. Blackenfelt, M. Managing Complexity by Product Modularisation: Balancing the Aspects of Technology and Business During the Design Process. Ph.D. Dissertation, Royal Institute of Technology, Stockholm, Sweden, 2001.
42. Nagel, K.; Erben, R.F.; Piller, F.T. (Eds.) *Produktionswirtschaft 2000: Perspektiven für die Fabrik der Zukunft*; Gabler Verlag: Wiesbaden, Germany, 1999; ISBN 9783322894823.
43. Göpfert, J.; Steinbrecher, M. Modulare Produktentwicklung leistet mehr. *Harv.-Bus.-Manag.* **2000**, 22, 20–30.
44. Seidel, M. *Methodische Produktplanung: Grundlagen, Systematik und Anwendung im Produktentstehungsprozess*; Universität Karlsruhe: Karlsruhe, Germany, 2005.
45. Jonas, H.; Gebhardt, N.; Krause, D. Towards a Strategic Development of Modular Product Programs. In *Proceedings of the 12th International Design Conference, Dubrovnik, Croatia, 21–24 May 2012*; pp. 959–968.
46. Hoisl, B. *Produkte Digital-First Denken*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2019; ISBN 978-3-658-23050-0.
47. Broy, M. Challenges in Automotive Software Engineering. In *Proceedings of the 28th International Conference on Software Engineering, Shanghai, China, 20–28 May 2006*; Association for Computing Machinery: New York, NY, USA, 2006; pp. 33–42, ISBN 978-1-59593-375-1.
48. Rodríguez, P.; Haghighatkah, A.; Lwakatare, L.E.; Teppola, S.; Suomalainen, T.; Eskeli, J.; Karvonen, T.; Kuvaja, P.; Verner, J.M.; Oivo, M. Continuous Deployment of Software Intensive Products and Services: A Systematic Mapping Study. *J. Syst. Softw.* **2017**, 123, 263–291. [[CrossRef](#)]
49. Kizgin, U.V.; Stein, A.; Esapathi, J.; Vietor, T. Systematic Method for Identifying Safety and Security Requirements in Autonomous Driving: Case Study of Autonomous Intersection System. *Appl. Syst. Innov.* **2025**, 8, 168. [[CrossRef](#)]
50. Nolte, B.; Stein, A.; Vietor, T. Designing a Method for Identifying Functional Safety and Cybersecurity Requirements Utilizing Model-Based Systems Engineering. *Appl. Syst. Innov.* **2025**, 8, 45. [[CrossRef](#)]
51. Şahin, T.; Inkermann, D.; Vietor, T. Towards Consistent Value Orientation in Release Planning. In *Proceedings of the ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Anaheim, CA, USA, 18–21 August 2019*; American Society of Mechanical Engineers: Anaheim, CA, USA, 2019; ISBN 978-0-7918-5927-8.
52. Jantunen, S.; Lehtola, L.; Gause, D.C.; Dumdim, U.R.; Barnes, R.J. The Challenge of Release Planning. In *Proceedings of the 2011 5th International Workshop on Software Product Management (IWSPM), Trento, Italy, 30 August 2011*; pp. 36–45, ISBN 978-1-4577-1146-6.
53. Sommerville, I. *Software Engineering*, 9th ed.; Pearson: Boston, MA, USA, 2011; ISBN 9780137053469.
54. Schuh, G.; Eversheim, W. Release-Engineering—An Approach to Control Rising System-Complexity. *CIRP Ann.* **2004**, 53, 167–170. [[CrossRef](#)]
55. Kühn, A.; Bremer, C.; Dumitrescu, R.; Gausemeier, J. Feature Models Supporting Trade-off Decisions in Early Mechatronic Systems Design. In *Proceedings of NordDesign 2014 Conference, Espoo, Finland, 27–29 August 2014*; Laakso, M., Ed.; Aalto Design Factory, Aalto University: Aalto, Finland, 2014; pp. 835–844, ISBN 978-1-904670-59-9.
56. Greer, D.; Ruhe, G. Software Release Planning: An Evolutionary and Iterative Approach. *Inf. Softw. Technol.* **2004**, 46, 243–253. [[CrossRef](#)]
57. Inkermann, D.; Huth, T.; Vietor, T. Towards Cross-domain Release Engineering: Potentials and Challenges for Automotive Industry. In *ADAPTIVE 2018: The Tenth International Conference on Adaptive and Self-Adaptive Systems and Applications*; Rausch, A., Knieke, C., Schranz, M., Eds.; IARIA: Barcelona, Spain, 2018; pp. 93–99, ISBN 9781612086101.
58. Wright, H.K. *Release Engineering Processes, Their Faults and Failures*; University of Texas: Austin, TX, USA, 2012.
59. Şahin, T.; Huth, T.; Axmann, J.; Vietor, T. A Methodology for Value-oriented Strategic Release Planning to Provide Continuous Product Upgrading. In *Proceedings of the 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 14–17 December 2020*; p. 5.
60. Sudhoff, W. *Methodik zur Bewertung Standortübergreifender Mobilität in der Produktion*; H. Utz: München, Germany, 2008; ISBN 9783831607495.

61. Patzak, G. *Systemtechnik: Planung Komplexer Innovativer Systeme. Grundlagen, Methoden, Techniken*; Springer: Berlin/Heidelberg, Germany, 1982; ISBN 978-3-540-11783-4.
62. Braun, S.; Diehl, H.; Petermann, M.; Hellenbrand, D.; Lindemann, U. Function Driven Process Design for the Development of Mechatronic Systems. In Proceedings of the 9th International Design Structure Matrix Conference, Munich, Germany, 16–18 October 2007; pp. 161–163.
63. Apel, S.; Batory, D.; Kästner, C.; Saake, G. *Feature-Oriented Software Product Lines*; Springer: Berlin/Heidelberg, Germany, 2013; ISBN 978-3-642-37520-0.
64. Kang, K.C.; Cohen, S.G.; Hess, J.A.; Novak, W.E.; Peterson, A.S. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*; Defense Technical Information Center: Fort Belvoir, VA, USA, 1990.
65. Botterweck, G.; Pleuss, A.; Polzer, A.; Kowalewski, S. EvoFM: Feature-driven Planning of Product-line Evolution. In *FOSD '09: Proceedings of the First International Workshop on Feature-Oriented Software Development, Denver, CO, USA, 6 October 2009*; ACM: New York, NY, USA, 2010; p. 168, ISBN 9781605589688.
66. Lindgren, M. *Towards a Capability Model for Release Planning of Software Intensive Systems*; Mälardalen University, School of Innovation, Design and Engineering: Eskilstuna and Västerås, Germany, 2008.
67. Şahin, T. *Continuous Upgrading of Complex Products by Value-Oriented Strategic Release Planning*; Technische Universität Braunschweig: Braunschweig, Germany, 2022.
68. Baumann, U. Mercedes-AMG E63 S Final Edition. V8-Abgesang hat Seinen Preis, Limo Bereits Ausverkauft. Available online: <https://www.auto-motor-und-sport.de/neuheiten/mercedes-amg-e63-s-final-edition/> (accessed on 5 November 2022).
69. Sommer, M. Rückruf Porsche 911 GT3: Supersportler hat 'ne Schraube Locker. Available online: <https://www.auto-motor-und-sport.de/verkehr/rueckruf-porsche-911-gt3-sicherheitsgurt-probleme/> (accessed on 6 November 2022).
70. Zhang, Z.; Lou, Q. A grey measurement of product complexity. In Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics, Montréal, QC, Canada, 7–10 October 2007; Zhang, Z., Luo, Q., Eds.; Omni Press: Montreal, QC, Canada, 2007.
71. Orfi, N.; Terpenney, J.; Sahin-Sariisik, A. Harnessing Product Complexity: Step 1—Establishing Product Complexity Dimensions and Indicators. *Eng. Econ.* **2011**, *56*, 59–79. [\[CrossRef\]](#)
72. Şahin, T.; Köster, L.; Huth, T.; Vietor, T. How to Upgrade Vehicles? Release Planning in the Automotive Industry. In *21. Internationales Stuttgarter Symposium*; Bargende, M., Reuss, H.-C., Wagner, A., Eds.; Springer: Wiesbaden/Heidelberg, Germany, 2021; pp. 155–173, ISBN 9783658335205.
73. Welge, M.K.; Al-Laham, A. *Strategisches Management: Grundlagen, Prozess, Implementierung*; 4th aktualisierte Aufl.; Gabler Verlag: Wiesbaden, Germany, 2003; ISBN 9783322997487.
74. Larman, C.; Basili, V.R. Iterative and Incremental Developments: A Brief History. *Computer* **2003**, *36*, 47–56. [\[CrossRef\]](#)
75. Blessing, L.; Chakrabarti, A. *DRM: A Design Research Methodology*; Springer: London, UK, 2009; ISBN 978-1-84882-586-4. [\[CrossRef\]](#)
76. Schuh, G. *Effizienter Innovieren mit Produktbaukästen: Studienergebnisse und Leitfaden—Ein Beitrag zu Lean-Innovation*; WZL: Aachen, Germany, 2010; ISBN 978-3-926690-24-1.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.