August 2025

Bridging the Gap: Low-Code Platforms and the Future of ERP Customization

Adrian Abendroth
Chair of Business Informatics, esp. Processes and Systems, adrian.abendroth@lswi.de

Benedict Bender
Weizenbaum Institute for the Networked Society, benedict.bender@uni-potsdam.de

Bridging the Gap: Low-Code Platforms and the Future of ERP Customization

Completed Research Full Paper

Adrian Abendroth
University of Potsdam
Adrian.abendroth@wi.uni-potsdam.de

Benedict Bender
Weizenbaum Institute for the Networked Society
Benedict.bender@uni-potsdam.de

Abstract

Enterprise Resource Planning (ERP) systems are critical for business process integration but often lack the flexibility needed for rapid adaptation. Low-Code Platforms (LCP) have emerged as a solution, enabling both developers and non-developers to customize ERP systems through visual development tools, modular add-ons, and API-driven integrations. This study explores the ecosystem of ERP enhancements via LCPs, analyzing the goals, challenges, requirements, and use cases from the perspective of ERP-, LCP vendors, users, IT consultancies, and third-party developers. Based on a qualitative interview study, findings show that ERP vendors expand platform-based business models, while LCP vendors focus on scalability and reusability. Users leverage LCPs for cost-effective customization, but governance, licensing complexity, and integration challenges persists. The study contributes to platform ecosystem theory, enterprise agility, and open innovation research, emphasizing the role of LCPs in reshaping ERP ecosystems. Future research should investigate quantitative performance impacts, vendor strategies, and governance frameworks.

Keywords

Enterprise Resource Planning (ERP) Systems, Low-Code Platforms (LCP), Customization, Ecosystems

Introduction

Enterprise systems (ES), particularly Enterprise Resource Planning (ERP) systems, play a critical role in organizations by integrating business processes and enabling the flow of information (Meiryani et al., 2021). While traditional ERP systems are robust, they often lack the agility required to adapt to evolving business needs (Teece et al., 2016). Customization has historically been costly, time-consuming, and dependent on programming expertise, making it a significant barrier to ERP flexibility (Haddara et al., 2022). As organizations face increasing pressure to rapidly adapt business applications, alternative approaches to ERP customization are needed.

Low-code platforms (LCPs) have emerged as a potential solution, offering a fundamentally new way to modify ERP systems-one that has not previously been available in the ERP customization space. These platforms allow users to develop applications using visual and declarative tools, significantly reducing the reliance on traditional coding (Bock and Frank, 2021). LCPs enable API-driven ERP extensions, provide pre-built templates, and support non-technical user customization, bridging the gap between rigid ERP architectures and the demand for adaptable enterprise software (Hirzel, 2021; Picek, 2023; Rymer, 2017). LCPs are already being used for a variety of enterprise applications, ranging from simple dashboards (Bies et al., 2022) to complex workflow automation (Alamin et al., 2023). They also integrate with add-on marketplaces such as Microsoft AppSource, which offers thousands of extensions to enhance ERP functionality (Staub et al., 2021; Lourenço et al., 2023). While LCPs represent a promising avenue for ERP customization, their suitability and effectiveness in this context have not been systematically evaluated.

Despite growing adoption (Käss et al., 2023), little research has examined whether LCPs truly deliver on the promise of making ERP customization more flexible, cost-effective, and sustainable. The impact of LCP adoption from different stakeholder perspectives - such as ERP vendors, end users, and third-party

developers - remains underexplored (Naqvi and Drews, 2024). It is unclear whether LCPs will serve as a complementary or disruptive force in ERP ecosystems, and what challenges arise in their adoption.

This study evaluates the role of LCPs in ERP transformation addressing these two research questions (RQ):

- **RQ1:** How does the ecosystem of ERP enhancement through LCP function and how do stakeholders interact in it?
- **RQ2:** What are the goals, challenges, requirements, and use cases for ERP enhancement from the perspective of different stakeholders?

To explore these questions, we employ a qualitative research approach, conducting interviews with ERP vendors, LCP users, and IT consulting firms specializing in ERP integration. By empirically examining the suitability and adoption of LCPs in ERP environments, this study contributes to a deeper understanding of ES adaptability and the evolving role of low-code solutions in digital transformation.

Background

While ERP systems provide standardized functionalities, they often lack the flexibility required to accommodate unique business needs (Kerr & Houghton, 2014). Since many companies rely on identical ERP solutions as their competitors, differentiation becomes challenging. Customization is often necessary to align ERP systems with specific business processes and create competitive advantages (Balint, 2017). However, ERP customization remains complex and resource-intensive due to intricate data models and proprietary code structures. Customization efforts require specialist knowledge, making modifications costly and time-consuming (Benders et al., 2006). Additionally, vendor-imposed constraints often limit modification capabilities (Parthasarathy & Sharma, 2017), while the need for external consultants further increases expenses (Fryling, 2010; Hitt et al., 2002). The process is not only lengthy (Uppström et al., 2015; Rothenberger & Srite, 2009) and costly (Rothenberger & Srite, 2009) but also presents risks, as rigid ERP architectures may hinder companies from adapting to evolving market demands (Lokuge & Sedera, 2017).

To address these challenges, **Low-Code Platforms (LCP)** such as Mendix and OutSystems have emerged as flexible alternatives for ERP customization (Yan, 2021). Unlike traditional development, LCP solutions enable both developers and non-developers to create and modify business applications with minimal effort (Adrian et al., 2020; Bock & Frank, 2021). These platforms rely on visual and declarative techniques, such as drag-and-drop interfaces, allowing users to design workflows, business logic, and user interfaces without deep coding knowledge (Metrôlho et al., 2019; Sahay et al., 2020; Totterdale, 2018). LCPs facilitate faster and cost-effective ERP customization by empowering business analysts and process owners to independently adjust workflows (Binzer et al., 2025; Domański et al., 2023) and user interfaces (Alamin et al., 2021; Hollick, 2021). They also support modular and reusable extensions, enhancing adaptability while reducing development efforts (Lebens, 2021; Di Ruscio et al., 2022).

Modern ERP systems are increasingly evolving into **platform ecosystems** that integrate **third-party add-ons and extensions** from different providers on two- or multi-sided markets and hence delivering synergy effects (Wang, 2021). Marketplaces such as Microsoft AppSource provide thousands of ERP extensions, reducing the need for costly in-house development (Staub et al., 2021; Lourenço et al., 2023). Add-ons offer cost-effective solutions for business-specific needs, often delivering faster implementation and lower maintenance costs compared to custom-built modifications (Olleros, 2008; Bosch & Bosch-Sijtsema, 2010). However, not all business requirements can be met through off-the-shelf add-ons, leading to additional customization needs (Parthasarathy & Daneva, 2016). Furthermore, companies must manage license fees and administrative overhead when integrating third-party extensions (Mantena et al., 2010).

LCPs provide a strategic advantage by bridging the customization gap of legacy systems, enabling resource-efficient customization during ERP implementation and ongoing adaptation on operation (Abendroth et al., 2024; Picek, 2023). ERP customization using LCPs can take various forms, depending on the integration level required (Gode et al., 2023):

1. **LCP ERP Customizing** – Some modern ERP systems (e.g., SAP Build) offer **built-in low-code frameworks** to enhance existing processes or digitize additional workflows.

2. **LCP ERP Extension** – For legacy ERP systems lacking flexibility, **standalone low-code platforms** (e.g., Mendix) are used to extend ERP capabilities via API-based applications.
3. **Low-Code ERP** – In rare cases, entire ERP systems are **built or migrated** onto LCP solutions (e.g., Thinkwise), creating a fully customized alternative to traditional ERP systems.

Beyond technical advantages, LCP foster open innovation by enabling a broader range of employees to contribute to digital transformation. Studies show that intuitive digital tools empower employees to actively shape innovation processes (Laviolette et al., 2016), and many companies are leveraging co-development platforms to enhance knowledge sharing (Mueller & Renken, 2017; Di Rocco et al., 2015). As enterprises increasingly prioritize adaptability and innovation, LCPs are gaining attention from analyst firms, highlighting their growing role in enterprise software customization (Bratincevic, 2024; Gartner, 2022).

Methodology

To address the research questions regarding stakeholder perspectives in the LCP-ERP ecosystem, we conducted a qualitative empirical interview study and applied content analysis techniques following Mayring (2020). Structured content analysis was used to systematize findings, identify patterns, and derive categories from the interview material. Data collection and category building is described next.

Data Collection and Sample: To ensure comprehensive insights, we selected three key stakeholder groups: LCP vendors, LCP consultancies, and LCP users. Slightly adapted questionnaire guidelines were developed for each group to capture their unique perspectives, and an ERP vendor was included to contrast findings. The design of these guidelines followed the methodologies of Faulbaum et al. (2009) and Mayring (2020) covering four high-level categories: (i) goals and reasons for integrating LCP with ERP systems; (ii) challenges associated with LCP; (iii) requirements of LCP with respect to ERP systems; and (iv) use cases for LCP that enhance ERP systems. Using a semi-structured interview approach (Gläser & Laudel, 2010), we allowed interviewers flexibility to explore emerging themes while maintaining a structured framework (Döring & Bortz, 2016). Given the lack of prior empirical research in this area, 11 in-depth interviews were conducted in late 2021, providing initial insights into the research questions. Interviews lasted an average of 47:55 minutes and included participants from various industries and organizational roles (see Table 1).

ID	Type	Industry	Position	ID	Type	Industry	Position
I1	ERP vendor	Technology	Senior ERP developer	I7	User	Public sector	LCP developer
I2	LCP vendor	Technology	Head of business development	I8	User	Wholesale for sanitary	CIO
I3	LCP vendor	Technology	Director of marketing & developer	I9	User	Technology	Head of master data
I4	LCP consul-tancy	Technology	LCP developer	I10	User	Automotive	Digital program manager
I5	LCP consul-tancy	Technology	Business architect	I11	User	Pharma-ceutical	Digital program manager
I6	User	Technology	LCP developer				

Table 1. Overview of interview partners

Category building: Following Mayring's (2020) approach to qualitative content analysis, we employed structuring as the primary analytical technique. This method extracts a framework from transcribed interview material to categorize and synthesize key themes. A coding guide was developed to ensure consistency, containing definitions, anchor examples, and coding rules (Mayring, 2020). Initially, the

transcribed content was coded based on the four predefined categories. As the analysis progressed, new subcategories emerged inductively, refining the coding framework through an iterative process to enhance intercoder reliability. An example of the developed coding guidelines is provided in Table 2.

Subcategory	Description	Coding rule	Anchor example
Easy Maintain-ability	It refers to the ability to manage, update, and adapt applications across their LCM, covering flexibility, version control, tracking and modifying develop-ments within the platform.	Only statements explicitly mentioning maintainabi-lity as a goal or benefit are included. General flexibili-ty or adaptability without reference to long-term maintenance is not coded.	[…] and say it's more maintainable. It's quicker and it's flexible. It's manageable, you can achieve anything while keeping track of what's being done [I9].
Data security	It encompasses measures and requirements to protect sensitive information within ERP systems and associated platforms. This includes compliance with regulations (e.g., GDPR), tracking data access, and ensuring that private or restricted infor-mation is handled properly.	Statements explicitly referring to security concerns, regulatory com-pliance, or the need for controlled data handling are included. General mentions of system rel-iability or access manage-ment without a security focus are not coded.	The Datenschutzerlass in Europe and you need to be able to track everything but you can not register private sentence for example. If it is not supported by your ERP system you gonna need something else to do it, right? [I4]

Table 2. Exemplary coding guidelines, category "LCP requirements"

Results

Goals in the Ecosystem of Stakeholders: During the interviews, the goals as well as dependencies between different roles in the ecosystem of ERP systems and LCP became apparent. Overall, four major roles and one secondary role were identified. The main participants are ERP vendors, LCP vendors, users of both systems, and IT consultancies specializing in LCP integrations. Secondary contributors include third-party developers. ERP vendors aim to enhance their systems by extending functionality and providing predefined configurations reducing the need for supervising complex customization. They aim to increase automation and efficiency by offering ERP marketplaces for add-ons and integrating integrated low-code platforms ("LCP ERP customization") that enable process improvements and workflow digitization without extensive coding, generating extra license fees [I1]. In addition, vendors are seeking to ensure centralized lifecycle management (LCM) and governance within their LCPs, while leveraging AI-powered capabilities to suggest potential new applications, further optimizing user experience and system adaptability [I1].

Similarly, LCP vendors strive to facilitate rapid development [I3-4, I6, I8-10], particularly by offering a component library that enables users to reuse existing application modules [I6, I11]. One interviewee emphasized: "The only moment when you really gain speed in development is when you have reusable components. Simply using Mendix instead of Java or React does not inherently make development faster. However, being able to reuse past components does. You need at least 6–7 applications to justify adopting the platform" [I6]. Consequently, LCP vendors should offer a broad range of predefined components and libraries of apps to accelerate adoption and enhance efficiency, with license fees serving as a key incentive.

For users (companies), the primary motivation in utilizing both ERP and LCP solutions is process optimization, particularly for rapid adaptation to new circumstances and the ability to develop modular components for future use. Another major goal is the consolidation of existing systems [I4, I6, I10-11]. Companies approach this either by replacing outdated ERP systems with LCP-based applications ("Low-Code ERP") or by extending existing ERP systems using LCPs ("LCP ERP extension"). A key question that arose during the interviews was why companies do not simply switch to an ERP system with a natively integrated LCP solution instead of adopting Low-Code ERP customization or extensions. The reasons identified were that many legacy ERP systems do not offer integrated LCP capabilities and upgrading an

older ERP system to a new version that includes LCP functionality incurs significant license fees, consulting costs, and time investments. Consequently, many companies find it more cost-effective to either extend their existing ERP with LCP or fully transition to a Low-Code ERP [I3-4, I9]. Finally, IT consultancies play a crucial role in supporting organizations in integrating LCPs with their ERP systems. They typically operate as independent service providers, partnering with both LCP and ERP vendors or as subsidiaries of LCP vendors, specializing in ERP integrations based on experience [I4, I9].

Due to the diverse stakeholders within the ecosystem and the exchange of services through offered add-ons and customizations, ERP systems and LCPs are evolving into software ecosystems. They (i) offer tools for developers to create complementary software functionalities; and (ii) providing an application marketplace, where the developed functionalities can be presented to match supply and demand. Third-party developers can generate revenue by offering paid or free extensions, while users benefit from a broader selection of prebuilt enhancements (Bender, 2021). In this context, the third-party developer can display extensions for the ERP or LCP marketplace. This increases the portfolio of the software systems, which makes them more attractive to customers. In addition, LCP consultancies can offer integrations from customer projects on the marketplace to (i) create new revenue streams through extension sales; or (ii) get attention for new customer projects by offering free extensions on the marketplaces. This can then lead to new customer projects. LCP and ERP vendors also incorporate key functions of popular add-ons into the system core. The streams and dependencies of the LCP enhancement ecosystem are depicted in Figure 1.

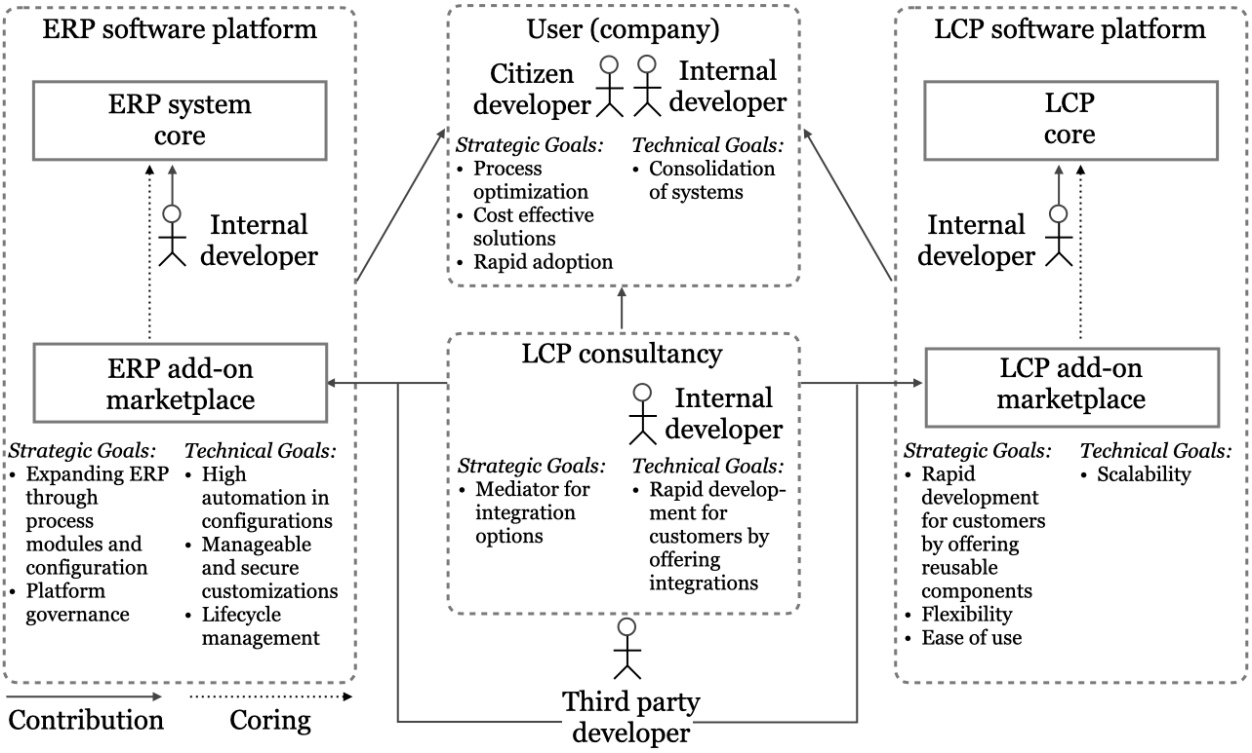

Figure 1. Ecosystem of LCP enhancement

Challenges of Stakeholders in ecosystem: The integration of LCPs with ERP systems presents challenges stemming from technical limitations, business model uncertainties, and ecosystem complexity. For users, a major limitation is the insufficient data analysis capabilities within LCPs [I2, I10]. Instead of relying on LCPs for analytical purposes, companies often resort to specialized Business Intelligence systems, which offer more advanced capabilities for reporting and data visualization. Another concern is the lack of transparency in LCP licensing models [I6, I10]. Companies struggle to understand whether costs are calculated per app, per user, per block, or per object, making it difficult to predict total expenses and evaluate the long-term financial impact of adopting an LCP. ERP vendors face legal and contractual challenges, particularly regarding custom code modifications made by customers using LCPs. Since

companies can independently develop extensions, vendors often refuse to assume liability for custom development, where customers implement their own modifications and custom logic [I1]. This creates uncertainty about support and warranty agreements, as vendors are reluctant to take responsibility for unintended side effects introduced by external modifications. For LCP vendors, one of the key challenges is defining the exact scope of projects that customers develop using their platform [I4]. Unlike traditional software development, where requirements are typically well-defined upfront, LCP projects tend to evolve dynamically. This can lead to uncertainties in project management, governance, and quality assurance. LCP consultancies also face significant challenges, particularly due to the high level of ERP knowledge required for successful integrations. Many ERP systems do not provide standardized endpoints for all business processes, requiring extensive customization to enable seamless interaction between LCP applications and core ERP functionalities [I9]. "The business cases are not always covered by standard endpoints. So you need a lot of ERP knowledge to actually implement the functionalities" [I9]. Additionally, as LCP and ERP vendors increasingly incorporate popular add-on functionalities into their system cores, third-party developers and LCP consultancies risk losing influence and revenue opportunities within the marketplaces.

Requirements of Stakeholders in ecosystem: Users prioritize maintainability, seeking solutions that are easy to update, modify, and scale [I10-11]. Additionally, flexibility in deployment options is a major concern. Companies expect support for different operational modes, allowing them to run LCP applications both in the cloud and on-premise based on their IT strategy and regulatory constraints [I10]. Another essential requirement is the accessibility of LCPs for Citizen Developers [I6, I10]. Since LCPs are often promoted to enable business users with limited coding expertise, companies expect possibility to enable business users for creating applications. For LCP vendors, meeting enterprise demands requires a comprehensive technology stack beyond core low-code features. Companies increasingly expect integrated capabilities in the whole LCM such as Robotic Process Automation, and Workflow Management Systems to streamline complex business processes [I3]. Built-in workflow and test engines are critical to handle end-to-end automation and maintain software quality [I4]. Software developers working with LCPs prioritize maintainability [I9], similar to users, as customizations from various departments could increase technical debt. Since LCPs are often used for rapid development, it is crucial that the generated applications remain modular, well-structured, and easy to debug or extend over time. Additionally, there is a growing demand for mobile app development support within LCPs [I5]. Many ERP-related workflows require mobile accessibility, and developers expect LCPs to offer native or responsive mobile solutions without extensive customization. Lastly, IT security is a core requirement [I9]. As LCPs are increasingly used to handle sensitive business data, developers expect robust security mechanisms, compliance with industry standards, and built-in features for access control and encryption.

Typical use cases for LCP enhancements in ERP systems: LCPs enhance ERP systems enabling faster development, seamless integration, and improved usability. One key use case is system integration, where LCPs simplify data exchange between ERP and other enterprise applications such as CRM, finance, and HR systems [I10]. Additionally, mobile access and forms allow real-time data entry and approvals through custom mobile applications, improving flexibility for users [I2]. LCPs also support prototyping, enabling rapid testing of new ERP functionalities before full implementation [I2]. For ERP vendors, a major benefit is user interface reusability, allowing modernization of interfaces without altering core system logic [I1]. Also, LCPs facilitate the development of CRUD applications, providing structured and user-friendly data management for ERP users [I3]. By streamlining customization and integration, LCPs make ERP systems more agile, accessible, and scalable, reducing development time while improving user experience.

Discussion

The ERP-LCP ecosystem consists of ERP vendors, LCP vendors, IT consultancies, users, and third-party developers, each playing a distinct role in enhancing ERP systems through LCP integration. ERP vendors focus on platform expansion by facilitating ERP and LCP add-on marketplaces while ensuring LCM and governance for customer customizations by offering native low-code tools for in-system customization (LCP ERP Customizing). LCP vendors enable agile ERP extensions by providing connectors, prebuilt templates, and migration tools, supporting both external ERP integrations (LCP ERP Extension) and full ERP replacements (Low-Code ERP). Companies leverage LCPs primarily for process optimization, system

consolidation, and customization, either by enhancing their ERP systems with internal LCP tools (LCP ERP Customizing) or building external applications that interact with legacy ERPs. IT consultancies act as intermediaries, offering expertise in LCP integration, add-on development, and migration strategies, while third-party developers expand ecosystem functionalities through marketplace-driven add-ons. This dynamic ecosystem fosters a shift toward modular, user-driven ERP adaptability, reducing reliance on high-code development and enabling more accessible enterprise innovation. Supplementary material on Open Science Framework (OSF) compares the three LCP customization models in ERP.[1] The goals of ERP vendors center on standardization, automation, and reducing customization maintenance, while LCP vendors focus on scalability and reusability. Users seek cost-effective flexibility, and LCP consultancies drive integration and business expansion. Key challenges include legal constraints, unclear licensing models, data analytics limitations, and governance issues in ERP modifications. To address these, stakeholders emphasize maintainability, hybrid deployment options (cloud/on-premises), modular architectures, and LCM. LCPs enhance ERP systems by enabling system integration, mobile access, UI modernization, and workflow automation, reducing development time and fostering a more adaptable, user-driven enterprise software landscape. Supplementary material on OSF discusses collaboration opportunities and potential conflicts between stakeholders.

Implications for Practice: *ERP vendors* can leverage LCP integration as an opportunity for new business models, particularly by expanding ERP marketplaces with LCP-based add-ons. However, they must address liability concerns surrounding user-built customizations. To mitigate risks, vendors could introduce certification programs for third-party extensions, ensuring compliance with security, performance, and governance standards. Embedding native low-code capabilities within ERP systems could provide a competitive advantage, allowing vendors to maintain greater control over customizations while reducing reliance on external LCP providers. *For companies*, LCPs enable greater flexibility and reduce reliance on IT departments, but proper governance is crucial to prevent uncontrolled modifications. Governance must cover technical facets like system integration and stability, along with organizational responsibilities and oversight. Implementing role-based access, IT oversight, and sandbox testing environments can help maintain system stability. Furthermore, licensing complexity remains a challenge, as LCPs vary in pricing models (e.g., per user, per transaction, or feature-based), making cost predictability difficult. Long-term maintainability is vital as frequent ERP updates or LCP platform changes may require continuous adaptation of custom solutions. For *IT consultancies*, the increasing demand for LCP-ERP integration expertise creates new service opportunities, particularly in industries with high customization needs, such as manufacturing, healthcare, and finance. Consultancies can differentiate themselves by specializing in tailored LCP add-ons for specific business processes, addressing the need for highly individualized ERP customizations. However, ERP systems vary significantly in API openness, with some offering standardized endpoints, while others require extensive custom integration efforts. Standardizing best practices for API-based LCP integrations will be key for consultancies to scale their services efficiently.

Theoretical Contributions: *First*, we extend platform ecosystem theory (Gawer & Cusumano, 2014) by illustrating ERP vendors' shift from monolithic systems to modular, multi-sided platforms through LCP-based customizations, raising governance challenges as vendors balance openness and control (Tiwana et al., 2010), ensuring quality, security, and LCM for third-party and user-developed extensions. *Second*, we contribute to dynamic capabilities theory (Teece et al., 1997) by demonstrating how LCPs enhance ERP adaptability, allowing firms to reconfigure business processes with minimal coding, reducing IT reliance, and improving enterprise agility. *Third*, we build on open and user-driven innovation theories (von Hippel, 2005; Chesbrough, 2003), highlighting how LCPs democratize ERP development by empowering non-technical users ("citizen developers") to create extensions, shifting software innovation from IT specialists to business units. However, this shift introduces new governance and integration challenges, as ERP vendors must oversee decentralized development while ensuring system coherency. *Finally*, we differentiate between three platform models—LCP ERP Customizing, LCP ERP Extension, and Low-Code ERP—each offering distinct levels of vendor control and ecosystem openness. The first model embeds LCPs within ERP environments, granting vendors strict governance while allowing controlled customization. The second relies on external LCP extensions, fostering open innovation but increasing integration complexity.

1 https://osf.io/wtv8k?view_only=3be96a9c69f64aed82acffc4cbebba29

The third, Low-Code ERP, replaces traditional ERP with a fully configurable low-code platform, providing maximum adaptability but limiting interoperability with existing ERP ecosystems. By linking platform structures to governance mechanisms, this study provides a nuanced understanding of stakeholder interactions, strategic trade-offs, and the evolving role of LCPs in enterprise software ecosystems.

Limitations: This study is based on qualitative data, which provides insights but limits generalizability across industries and geographic regions. Additionally, the limited number of interviews may not fully capture sector-specific adoption trends, and further studies could explore regional variations in LCP-ERP integration approaches. Furthermore, the fast-evolving nature of LCP and ERP ecosystems may render some findings outdated as vendors introduce new features, integration capabilities, and pricing models.

Future research should explore quantitative performance metrics to assess the impact of LCP-based ERP enhancements on efficiency, cost reduction, and system adaptability. Investigating barriers to user adoption, like training challenges, IT resistance, and organizational inertia, could provide deeper insights into LCP acceptance in enterprises. Variation across industries and organizational types should also be examined, as regulatory requirements and operational needs can significantly influence the perception and adoption of LCP-based ERP customizations. Another important avenue is analyzing ERP vendors' strategic responses to the growing LCP trend—will they embrace LCP integration, develop native solutions, or position themselves in competition with third-party platforms? Further research should also examine the long-term sustainability of LCP-driven ERP customizations—whether organizations continue using low-code modifications over time or revert to traditional IT-driven approaches. Lastly, governance and security challenges in LCP-based ERP environments require deeper investigation, particularly in terms of compliance, versioning, and LCM. This includes understanding how stakeholders navigate risks related to system stability, data privacy, and regulatory compliance when integrating LCPs into critical ERP systems.

Conclusion: This study highlights the transformative role of LCPs in ERP customization, addressing long-standing challenges in flexibility, cost, and accessibility. By defining the LCP-ERP ecosystem, we identified key stakeholder goals, challenges, and use cases, showing how LCPs enable agile enterprise adaptations through modular customization. While LCP adoption reduces IT dependency and fosters innovation, it introduces new complexities in governance, integration, and long-term maintainability. ERP vendors are increasingly integrating LCP-based customization models, either by expanding marketplaces or embedding native low-code functionalities. Future research should show how LCP-driven customizations affect the long-term sustainability of ERP systems, modularity, and the organizational agility of adopting companies. The study's findings contribute to platform ecosystem theory, enterprise agility, open innovation, and technology adoption research, offering a foundation for future studies on quantifying LCP impacts, vendor strategies, and governance challenges. As ERP ecosystems evolve, LCPs will play a crucial role in shaping the next generation of enterprise software adaptability.

Acknowledgements

This research is part of the research project PidMo (19F1195) funded by the German Federal Ministry for Digital and Transport (BMDV).

REFERENCES

Abendroth, A., Bender, B., & Gronau, N. (2024). The Evolution of Original ERP Customization: A Systematic Literature Review of Technical Possibilities. Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024), 1, 17–27.

Adrian, B., Hinrichsen, S., & Nikolenko, A. (2020). App Development via Low-Code Programming as Part of Modern Industrial Engineering Education. In I. L. Nunes (Ed.), Advances in Human Factors and Systems Interaction (pp. 45–51). Springer International Publishing.

Alamin, M. A. A., Malakar, S., Uddin, G., Afroz, S., Haider, T. B., & Iqbal, A. (2021). An Empirical Study of Developer Discussions on Low-Code Software Development Challenges. 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), 46–57.

Alamin, M. A. A., Uddin, G., Malakar, S., Afroz, S., Haider, T., & Iqbal, A. (2023). Developer discussion topics on the adoption and barriers of low code software development platforms. Empirical Software Engineering, 28(1), 4.

Balint, B. (2017). Maximizing the Value of Packaged Software Customization: International Journal of Enterprise Information Systems, 13(1), 1-16.

Bender, B. (2021). Platform Coring on Digital Software Platforms. Springer Fachmedien Wiesbaden.

Benders, J., Batenburg, R., & van der Blonk, H. (2006). Sticking to standards; Technical and other isomorphic pressures in deploying ERP-systems. Information & Management, 43, 194–203.

Bies, L., Weber, M., Greff, T., & Werth, D. (2022). A mixed-methods study of low-code development platforms: Drivers of digital innovation in SMEs. Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 1–6.

Binzer, B., Fuerstenau, D., & Winkler, T. (2025). Bridging Business and IT Through Low-Code/No-Code: Insights into Business-IT Collaboration in Enterprise Citizen Developer Programs. Proceedings of the 58th Hawaii International Conference on System Sciences, 451–460.

Bock, A. C., & Frank, U. (2021). Low-Code Platform. Business & Information Systems Engineering, 63(6).

Bratincevic, J. (2024, January 29). The Low-Code Market Could Approach $50 Billion By 2028. Forrester. https://www.forrester.com/blogs/the-low-code-market-could-approach-50-billion-by-2028/

Bosch, J., & Bosch-Sijtsema, P. (2010). From integration to composition: On the impact of software product lines, global development and ecosystems. Journal of Systems and Software, 83(1), 67–76.

Chesbrough, H. (2003). The Logic of Open Innovation: Managing Intellectual Property. California Management Review, 45(3), 33–58.

Domański, R., Wojciechowski, H., Lewandowicz, J., & Hadaś, Ł. (2023). Digitalization of Management Processes in Small and Medium-Sized Enterprises—An Overview of Low-Code and No-Code Platforms. Applied Sciences, 13(24), Article 24.

Döring, N., & Bortz, J. (2016). Forschungsmethoden und Evaluation (5.th). Springerverlag.

Di Rocco, J., Di Ruscio, D., Iovino, L., & Pierantonio, A. (2015). Collaborative Repositories in Model-Driven Engineering [Software Technology]. IEEE Software, 32(3), 28–34. IEEE Software.

Di Ruscio, D., Kolovos, D., De Lara, J., Pierantonio, A., Tisi, M., & Wimmer, M. (2022). Low-code development and model-driven engineering: Two sides of the same coin? Software and Systems Modeling, 21(2), 437–446.

Faulbaum, F., Prüfer, P., & Rexroth, M. (2009). Was ist eine gute Frage? VS Verlag für Sozialwissenschaften.

Fryling, M. (2010). Estimating the impact of enterprise resource planning project management decisions on post-implementation maintenance costs: A case study using simulation modelling. Enterprise Information Systems, 4(4), 391–421.

Gartner. (2022, December 13). Gartner Forecasts Worldwide Low-Code Development Technologies Market to Grow 20% in 2023. https://www.gartner.com/en/newsroom/press-releases/2022-12-13-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-20-percent-in-2023

Gawer, A., & Cusumano, M. A. (2014). Industry Platforms and Ecosystem Innovation. Journal of Product Innovation Management, 31(3), 417–433.

Gläser, J., & Laudel, G. (2010). Experteninterviews und Qualitative Inhaltsanalyse. VS Verlag.

Gode, A., Roesner, D., Bingler, D., Naujoks, F., Sontow, K., & Finkler, M. (2023). Programmieren für Dummies: Bedeutet Low-Code das Ende von ERP? (p. 1-7). Bitkom e.V.

Haddara, M., Gøthesen, S., & Langseth, M. (2022). Challenges of Cloud-ERP Adoptions in SMEs. Procedia Computer Science, 196, 973–981.

Hitt, L. M., Wu, D. J., & Zhou, X. (2002). Investment in Enterprise Resource Planning: Business Impact and Productivity Measures. Journal of Management Information Systems, 19(1), 71–98.

Hirzel, M. (2023). Low-Code Programming Models. Communications of the ACM, 66(10), 76–85.

Hollick, R. (2021, September 2). How low-code / no-code can be used in ERP | ERP. SYSPRO Blog.

Käss, S., Strahringer, S., & Westner, M. (2023). Practitioners' Perceptions on the Adoption of Low Code Development Platforms. IEEE ACCESS, 11, 29009–29034.

Kerr, D., & Houghton, L. (2014). The dark side of ERP implementations: Narratives of domination, confusion and disruptive ambiguity. Prometheus, 32(3), 281–295.

Laviolette, E. M., Redien-Collot, R., & Teglborg, A.-C. (2016). Open innovation from the inside: Employee-driven innovation in support of absorptive capacity for inbound open innovation. The International Journal of Entrepreneurship and Innovation, 17(4), 228–239.

Lebens, M., Finnegan, R., & Sorsen, S. (2021). Rise of the Citizen Developer. Muma Business Review, 5(12), 101–111.

Lokuge, S., & Sedera, D. (2017). Turning dust to gold: How to increase inimitability of enterprise system. Proceedings of the 21st Pacific Asia Conference on Information Systems (PACIS 2017).

Lourenço, M., Gasiba, T. E., & Pinto-Albuquerque, M. (2023). You are doing it wrong: On vulnerabilities in low code development platforms. CYBER 2023: The Eighth International Conference on Cyber-Technologies and Cyber-Systems.

Mantena, R., Sankaranarayanan, R., & Viswanathan, S. (2010). Platform-based information goods: The economics of exclusivity. Decision Support Systems, 50(1), 79–92.

Mayring, P. (2020). Qualitative Inhaltsanalyse. In Handbuch qualitative Forschung in der Psychologie (pp. 495–511). Springer.

Meiryani, Fernando, E., Hendratno, S. P., Kriswanto, & Wifasari, S. (2021). Enterprise Resource Planning Systems: The Business Backbone. Proceedings of the 5th International Conference on E-Commerce, E-Business and E-Government, 43–48.

Metrôlho, J., Araújo, R., Ribeiro, F., & Castela, N. (2019). An approach using a low-code platform for retraining professionals to ICT. 11th International Conference on Education and New Learning Technologies, 7200–7207.

Mueller, B., & Renken, U. (2017). Helping Employees to be Digital Transformers – the Olympus.connect Case. Thirty-Eighth International Conference on Information Systems, Seoul.

Naqvi, S. A. A., & Drews, P. (2024). Understanding Low-Code Evolution, Adoption and Ecosystem for Software Development. The 15th International Conference on Software Business (ICSOB 2024).

Olleros, X. (2008). The lean core in digital platforms. Technovation, 28(5), 266–276.

Parthasarathy, S., & Daneva, M. (2016). An approach to estimation of degree of customization for ERP projects using prioritized requirements. Journal of Systems and Software, 117, 471–487.

Parthasarathy, S., & Sharma, S. (2017). Impact of customization over software quality in ERP projects: An empirical study. Software Quality Journal, 25(2), 581–598.

Picek, R. (2023). Low-code/No-code Platforms and Modern ERP Systems. 2023 International Conference on Information Management (ICIM), 44–49.

Rothenberger, M. A., & Srite, M. (2009). An Investigation of Customization in ERP System Implementations. IEEE Transactions on Engineering Management, 56(4), 663–676.

Rymer, J. R. (2017). The Forrester WaveTM: Low-Code Development Platforms For AD&D Pros, Q4 2017 (The Forrester Wave, pp. 1–21). Forrester.

Sahay, A., Indamutsa, A., Di Ruscio, D., & Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms. 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 171–178.

Staub, N., Haki, K., Geneva School of Business, Aier, S., University of St. Gallen, Winter, R., University of St. Gallen, Magan, A., & salesforce.com. (2021). Acquisition of Complementors as a Strategy for Evolving Digital Platform Ecosystems. MIS Quarterly Executive, 20(4), 237–258.

Teece, D. J., Pisano, G., & Shuen, A. (1997). Dynamic capabilities and strategic management. Strategic Management Journal, 18(7), 509–533.

Teece, D., Peteraf, M., & Leih, S. (2016). Dynamic Capabilities and Organizational Agility: Risk, Uncertainty, and Strategy in the Innovation Economy. California Management Review, 58(4), 13–35.

Totterdale, R. L. (2018). Case study: The utilization of low-code development technology to support research data collection. Issues in Information Systems, 19(2), Article 2.

Tiwana, A., Konsynski, B., & Bush, A. A. (2010). Research Commentary—Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. Information Systems Research, 21(4), 675–687.

Uppström, E., Lönn, C.-M., Hoffsten, M., & Thorström, J. (2015). New Implications for Customization of ERP Systems. 2015 48th Hawaii International Conference on System Sciences, 4220–4229.

Wang, P. (2021). Connecting the Parts with the Whole: Toward an Information Ecology Theory of Digital Innovation Ecosystems. MIS Quarterly, 45(1), 397–422.

Yan, Z. (2021). The Impacts of Low/No-Code Development on Digital Transformation and Software Development (arXiv:2112.14073; Issue arXiv:2112.14073). arXiv.

von Hippel, E. (2005). Democratizing Innovation. The MIT Press.