

RESEARCH ARTICLE

Practitioners' Perceptions on the Adoption of Low Code Development Platforms

SEBASTIAN KÄSS¹, SUSANNE STRAHINGER², AND MARKUS WESTNER¹¹Faculty of Computer Science and Mathematics, Ostbayerische Technische Hochschule (OTH) Regensburg, 93053 Regensburg, Germany²Faculty of Business and Economics, Technische Universität Dresden, 01062 Dresden, Germany

Corresponding author: Markus Westner (Markus.Westner@oth-regensburg.de)

This work was supported by the Ostbayerische Technische Hochschule (OTH) Regensburg.

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT Organizations are under increasing pressure to develop applications within budget and time at high quality. Therefore, multiple organizations adopt a Low Code Development Platform (LCDP) to develop applications faster and cheaper compared to traditional application development. However, current research on LCDP adoption lacks empirical grounding as well as a deeper understanding of the importance of adoption drivers and inhibitors. We conducted semi-structured interviews and a Delphi study with 17 experts to address these gaps. As a result, we identified 12 drivers and 19 inhibitors for adopting LCDPs. We show that the experts have a consensus on the most and the least important drivers and inhibitors for LCDP adoption. Yet, the ranking of the drivers and inhibitors between the most and least important is context-dependent. For some drivers and inhibitors, the experts' ranking is similar to academic literature, whereas, for others, it differs. In conclusion, the study at hand empirically validates drivers and inhibitors for LCDP adoption, adds six new drivers and six new inhibitors to the body of knowledge, and analyzes the importance of these factors.

INDEX TERMS Low code, Delphi study, software development, adoption, driver, inhibitor.

I. INTRODUCTION

Even before the COVID-19 pandemic, organizations felt competitive pressure to digitalize business models and internal processes [1]. Therefore, they must increase their speed in developing applications within budget and time constraints [2]. However, a significant market gap exists among skilled software developers for application development [3]. An option to react to these challenges is to use Low Code Development Platforms (LCDPs), which are promoted to increase efficiency, effectiveness, reduce costs, and empower users [4], [5], [6], [7], [8]. Moreover, LCDP vendors advertise their products as being capable of supporting professional software developers but also developers in the business department or regular business employees (often referred to as citizen developers) who develop applications with little to no programming experience [4], [5], [6], [9].

LCDPs have been a topic of discussion for multiple years among practitioners [4] before academia started to research

the topic [4], [10]. Gartner predicts that by “2023, over 50% of medium to large enterprises will have adopted” [11, p. 1] an LCDP. However, practitioners and researchers are criticized for overly optimistic views on LCDPs [4]. If organizations adopt LCDPs and the benefits are incurred, the adoption can be a source of competitive advantage [12]. However, adopting LCDPs can also induce risks, e.g., by inexperienced developers neglecting security standards when developing applications [13], [14].

Despite the potential widespread use, it is academically under-researched what drives or inhibits the adoption of LCDPs. A literature review by [12] summarizes inhibitors and drivers for LCDP adoption. However, the authors criticize that the current research lacks some commonly discussed adoption factors (e.g., top management support was not found as a driver or inhibitor), lacks empirical grounding, and the importance of identified drivers and inhibitors needs to be clarified [12] to steer attention to the most influential aspects. Technology adoption is a well-researched phenomenon in academia. Nevertheless, as the LCDP adoption research is in its infancy, it is unclear if the general factors from technology

The associate editor coordinating the review of this manuscript and approving it for publication was Justin Zhang.

adoption research also apply to the specific situation of LCDP adoption. Further, to be useful for practitioners, the research on drivers and inhibitors must be much more specific than the factors discussed in traditional technology adoption research. Therefore, the study at hand addresses the following research questions (RQ):

RQ1: What are drivers and inhibitors of LCDP adoption?

RQ2: What is the importance of these drivers and inhibitors for LCDP adoption?

As the units of analysis, we select work systems, i.e., “systems in which human participants and/or machines perform work [...] using information, technology, and other resources” [15, p. 75]. Specifically, we focus on work systems where professional and citizen developers use LCDPs as information systems to carry out low code development. This perspective helps us to research the adoption on a level where it usually occurs - between the individual developer and an organizational level. To answer RQ1, we conducted explorative, qualitative semi-structured interviews. A subsequent ranking-type Delphi study with 17 experts answers RQ2. We followed the methodological guidelines of [16] for the Delphi study, with the analytical extension of best/worst scaling by [17], [18]. We chose a Delphi study as it allows us to create a consensus (on the importance) in the exploratory field of LCDP adoption, where we only have limited empirical evidence [16]. Moreover, the Delphi method allows us to quantitatively determine the quality of the consensus [16].

The study at hand has three contributions. First, we explicitly answer the calls for in-depth research on LCDPs [10], [12] and their adoption [12] empirically. We identify 12 drivers and 19 inhibitors for LCDP adoption and add six new drivers and six new inhibitors to the body of knowledge. Second, this study is the first to empirically research the importance of the drivers and inhibitors for LCDP adoption. This steers attention to the most critical aspects and allows practitioners to improve decision-making [19]. We can show that the experts strongly agree on the most and least important drivers and inhibitors, whereas the consensus is weaker for the drivers and inhibitors in between. Third, we show that the importance of drivers and inhibitors ranked on the basis of the LCDP adoption literature differs from the experts' results.

This study is structured as follows: in section two, we outline the conceptual background of LCDPs and present the current state of research on drivers and inhibitors of LCDP adoption. Section three presents the methodology, and the results are shown in section four. In section five, we discuss the results considering previous literature. The study concludes with the contributions, limitations, and a presentation of future research directions.

II. CONCEPTUAL BACKGROUND

A. LCDPs—CONCEPTUAL BACKGROUND

For many years, the discussion on Low Code Development (LCD) was mainly driven by practitioners [4], [20]. The first description of LCD came from Forrester Research,

characterizing it as software development with minimal source code through interactive graphical interfaces [5]. For the development, so-called LCDPs are used, which are “products and/or cloud services for application development that employ visual, declarative techniques instead of programming and are available to customers at low- or no-cost in money and training time to begin” [21, p. 4]. Moreover, LCDPs usually use the Platform as a Service (PaaS) delivery model [10], [21], [22].

LCDPs cannot be considered an entirely new phenomenon [4], [6], [23]. Instead, LCDPs draw from and combine previously well-known concepts, such as rapid application development, fourth-generation programming languages, model-driven development, and computer-aided software engineering [4], [6], [24], [25], [26]. Interestingly, numerous well-known LCDPs existed before the term was coined by Forrester Research and were re-branded after the term emerged [4].

As outlined by [6], the term LCDP is currently used to describe a list of heterogeneous development platforms with different technical capabilities, multiple scenarios of use, and different target audiences. This study focuses on LCDPs used by professional and citizen developers to develop applications within organizations. Gartner states that these platforms use “model-driven or visual development paradigms supported by expression languages and possibly scripting to address use cases such as citizen development, business unit IT, enterprise business processes, [or] composable applications” [11, p. 1]. Moreover, these LCDPs must include LCD capabilities (i.e., model-driven and graphical programming models) and support the development of applications consisting of a user interface, business logic, workflow, and data services [11]. Due to the expected benefits of LCDPs, an increasing number of organizations adopt them, and major IT vendors and multiple start-ups offer them [27]. Leading vendors in this field are Appian, Mendix, Microsoft, OutSystems, Salesforce, and ServiceNow [11].

A closely related concept is that of no code development platforms (NCDPs), with the difference that low code reduces hand coding, whereas no code eliminates it [28]. Researchers disagree on whether NCDPs are a different concept or part of the LCDP concept. Some authors consider it a different concept due to different functionality and scalability [29]. However, others see NCDPs primarily as a marketing statement [10], [29]. For this study, we follow the latter argumentation.

B. CURRENT STATE OF RESEARCH ON LCDP ADOPTION

We define LCDP adoption as “the first use or acceptance of” [30, p. 24] an LCDP within a work system. Technology adoption can occur on multiple levels, with the individual and organizational levels being the two extreme points [31]. For this study, we take a work system view, which helps us to explain the adoption that takes place between the individual and organizational levels [32], e.g., a group of developers decides to use an LCDP for a project. Drivers are factors that

facilitate the adoption, and inhibitors are factors that hinder the adoption [33] of LCDPs.

Current academic research on drivers and inhibitors for LCDP adoption is in its infancy [12]. A recent literature review by [12] applied the diffusion of innovation framework to synthesize the academic discussions on drivers and inhibitors for LCDP adoption. The authors found improved software development efficiency, reduced entry barriers for application development, and reduced required knowledge for application development as the most discussed drivers in academic literature [12]. In contrast, lack of governance, flexibility, customization, scalability, and limited portability significantly inhibit LCDP adoption [12]. Moreover, [12] conclude that research on LCDPs lacks substantial empirical evidence, as only two publications research LCDP adoption empirically, i.e., [34] and [35]. [35] examine the adoption on an individual level by analyzing posts in online forums to retrieve drivers and inhibitors for the adoption on an individual level. They point out that faster development (i.e., higher efficiency), ease of use, and lower complexity are the most often discussed drivers. The most discussed inhibitors are difficulties in learning LCDPs, high prices, and a lack of customization [35]. A survey among IT experts found that the main reasons to use LCDPs are accelerating digital transformation and reducing dependency on IT developers [34]. In contrast, concern about vendor lock-in, lack of knowledge about LCDPs, and lack of use cases are the main reasons for not using LCDPs [34].

Moreover, [22] posits that not only a single factor drives or inhibits the adoption of LCDPs. Instead, the authors argue that adopting LCDPs can only be researched by combining different factors that influence each other. Therefore, [22] builds a configurational research model to explain LCDP adoption. As theoretical lenses, the Technology-Organization-Environment (TOE) model and socio-technical systems (STS) theory are used [22] to combine and organize adoption factors derived from adoption literature in the fields of cloud computing and agile software development methods. However, the authors do not empirically validate their model but outline to do so as a next step.

For this paper, we take a similar approach and use a combination of the STS theory and TOE model to structure our drivers and inhibitors for LCDP adoption. [36] have already applied the STS theory to analyze computer-aided software engineering adoption, which is conceptually similar to LCDP adoption [25], [26]. In the context of LCDPs, the four categories of the STS theory have been defined by [32] as follows. The *structure* category is the nature of an organization's communication, authority, and setup [32]. The *people* category comprises all stakeholder-specific drivers and inhibitors [32]. The *task* category includes all application development-specific drivers and inhibitors, whereas the *technology* category includes all platform-specific issues [32]. However, the STS theory assumes that work systems must be open and responsive to the environment [37]. This aspect of work

systems is essential as it implies that LCDP adoption is also affected by environmental factors. Hence, we extend the four categories from the STS theory by the *environment* category, as indicated in the TOE model. The environment category then comprises all external drivers and inhibitors affecting LCDP adoption [38], [39].

III. METHODOLOGY

A. OVERVIEW

This study focuses on identifying (RQ1) and ranking (RQ2) drivers and inhibitors of LCDP adoption. We conducted semi-structured interviews and a ranking-type Delphi study to answer the research questions. Ranking-type Delphi studies are used to reach a group consensus about the relative importance of a set of factors and have seen widespread use in research [16], [41]. As outlined by [16], [17], [42], and [43], a Delphi study provides controlled feedback to the experts and usually consists of the phases expert selection, iterative data collection and data analysis, and data presentation. After our final Delphi data collection round, we added one additional round of interviews to discuss the results with selected experts. Discussing results before publishing is good practice in case study methodology, as it reduces the risk of misinterpreting the results and thus leads to a higher construct validity [44], [45], [46]. Therefore, we also decided to perform this step for the Delphi study's results, i.e., ultimately ending with five methodological steps. An overview of the extended methodology can be found in Fig. 1.

In each phase, certain design decisions have to be made – one of them is the mechanism to determine the ranking of the identified items. The literature discusses several mechanisms (i.e., direct ranking of items, ratings on pre-defined scales (e.g., Likert scales), allocation of points from a pre-defined pool) [17], [18], [47]. However, ties among items, standardization difficulties, or response-style biases are well-discussed limitations of these mechanisms [48]. Therefore, [17] and [18] propose using best/worst scaling as a ranking mechanism to overcome these limitations, as it “forces participants to discriminate between items by choosing the most distinct pair (i.e., participants do not have to assign discrete values to each item)” [17, p. 61]. Moreover, best/worst scaling is an easy-to-conduct and time-efficient way to collect empirical information in ranking-type Delphi studies [17]. Therefore, this study uses the approach of [16], with the analytical extension for ranking-type best/worst scaling outlined in [17] and [18].

B. EXPERT SELECTION

Selecting the right experts is crucial for Delphi studies, as the results mainly depend on a small number of selected experts [17], [43]. To choose appropriate experts, we applied the following approach, as outlined in [16] and [17]: identify expert groups, identify experts, nominate additional experts, rank experts, and invite experts.

As we aim to incorporate the perspective of experts with significant practical experience in LCDP adoption, we

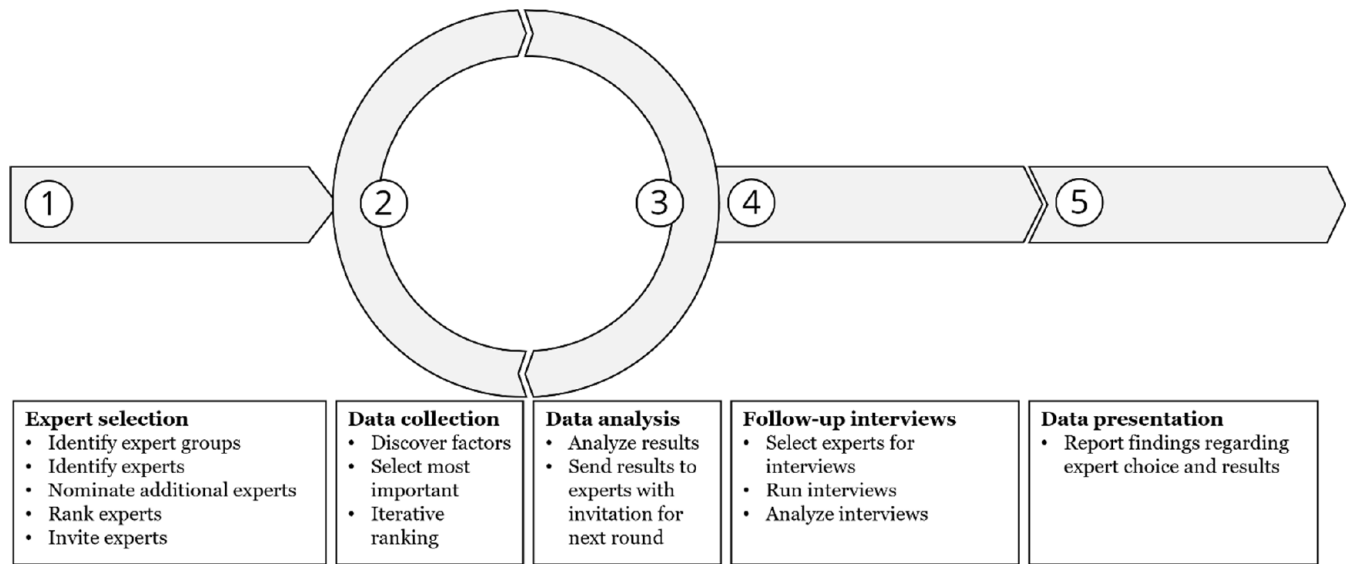


FIGURE 1. Methodology of the study based on [16], [17], [18] extended with follow-up interviews.

identified three expert groups: (1) consultants, (2) line managers, and (3) sales executives of LCDPs. The rationale for inviting consultants is that they support multiple organizations in adopting LCDPs and can therefore provide various perspectives on drivers and inhibitors for LCDP adoption. Line managers were selected, as they have in-depth expertise in LCDP adoption within one organization. Moreover, sales executives of LCDP vendors were picked, as they are in constant discussions with multiple customers who want to adopt LCDPs. Experts of all three groups must either be the decision-maker for LCDP adoption in a work system (line managers) or advise the decision-maker for the LCDP adoption in such a work system (consultants and sales executives).

We identified the experts through our professional network (as outlined by [16] and [17]) and through a search on the career portal LinkedIn. For our professional network, we contacted experts who either are the decision-makers or advise the decision-makers and invited them to participate in our research. Further, we asked the experts to act as gatekeepers (i.e., influential persons who can connect the researcher to additional experts [49]). On the career portal LinkedIn, we searched for “Low Code” as a search string, evaluated if the persons matched our expert requirements, and then invited them to participate in our research.

Through discussions with the experts, we could nominate additional experts. Through conversations with them, two additional experts were added to the initial list of 27 experts, leading to 29. Although there is still no agreement on the optimal number of experts for ranking-type Delphi studies [16], [50], the literature agrees that the number of experts should be manageable to reach a consensus. A typical panel size seems to be between seven and 30 [16]. As 29 is within this range, we invited all experts.

C. DATA COLLECTION AND DATA ANALYSIS

The Delphi method “repeatedly collects, analyzes, and reconciles data with experts.” [17, p. 63]. The data collection consists of (1) the discovery of factors, (2) the selection of the most important factors, and (3) the iterative ranking of the factors [17].

To discover the factors, we conducted semi-structured interviews with the experts. We decided to use interviews, as they allow the experts to list the drivers and inhibitors [16]. The interviews took place in March-May 2022, lasted between 19:55 to 56:06 minutes, and were recorded and transcribed. The interviews were structured around five themes: (1) the background of the interviewees, (2) the current state of LCDP adoption, (3) the observed process of LCDP adoption, (4) drivers for LCDP adoption, and (5) inhibitors for LCDP adoption. We analyzed the data through open and axial coding based on the guidelines of [51]. After we had extracted the initial list of factors, we checked for duplicates and consolidated the factors where possible. Furthermore, we created a description for each factor based on interview input.

To avoid overwhelming experts with many factors, academic literature [16] and [17] proposes selecting the most important factors and considers approximately 20 factors as the upper limit [42]. As we discovered 12 drivers and 19 inhibitors for LCDP adoption, we did not need to further reduce the number of factors. Before starting with the iterative ranking of factors, we followed [16] and defined three stop criteria: (1) Kendall’s $W > 0.7$, indicating a strong consensus, (2) three rounds had been run, or (3) no significant difference in the mean rank between two successive rounds. Moreover, we validated that all drivers and inhibitors from the literature review of [12] were part of the factors in the Delphi study.

For the iterative ranking of the factors, the survey must follow the design principles of frequency balance,

TABLE 1. Overview of participation rate throughout the multiple Delphi rounds.

Group	Interviews	Round 1		Round 2		Round 3		Follow-up
	Participated	Invited	Participated	Invited	Participated	Invited	Participated	Participated
All responses	15	17	14	17	12	17	8	4
Consultants	9	9	7	9	6	9	3	2
Line managers	3	5	5	5	4	5	4	2
Sales executives	3	3	2	3	2	3	1	0

orthogonality, connectivity, and positional balance [17], [47]. While it is possible to design the scaling manually, we used dedicated statistical software (Lighthouse Studio 9.14.1) to design and run the best-worst scaling survey, as proposed by [52]. To conduct the survey, the experts received an individual link to a web-based survey with 31 questions – the first 19 focused on inhibitors of LCDP adoption, the second 12 on drivers for LCDP adoption. Each question offered the experts five inhibitors (drivers) at a time, from which they chose the most important and the least important inhibitor (driver) (see Appendix A). After each round, we analyzed the data and sent the outcome, its interpretation, and the next round's survey to the experts.

A critical part of all Delphi studies is to motivate the experts to participate through multiple rounds [16], [53]. For this study, we sent numerous reminders to all experts and offered to donate to a charity for each participating expert. In this light, it is crucial to discuss the handling of experts skipping one round of the Delphi study and dropouts [54], [55]. There is no clear methodological guidance on this topic yet. Some studies (e.g., [56]) exclude the experts who did not participate in a round for the next round; others make a case-by-case decision (e.g., [54]), whereas others argue that experts can skip rounds and re-join in a later round of the Delphi study (e.g., [55]). The latter argue that they can skip and re-join in order to have a broad set of opinions [55]. We also followed the latter argumentation, as we wanted to have a broad set of opinions for our Delphi study. The results of each Delphi round (i.e., Kendall's W, best/worst score, mean rank, top-half rank, and trend) were shared with all invited experts to bring all experts to the same level of knowledge, even if they did not participate in the round. We ran three ranking rounds with the experts from June-September 2022. Each ranking round was open for four weeks, and we sent multiple reminders to the experts.

D. FOLLOW-UP INTERVIEWS

After the final ranking round, we decided to run additional follow-up interviews with selected experts to interpret and discuss the results of the Delphi study. We selected the experts with the largest Euclidian distance between their individual ranking and the mean ranking of all experts. The interviews

took place in October 2022 and lasted between 23:10 to 34:40 minutes. The interviews were centered around three themes: (1) discussion of changes since the last interview (e.g., progress of LCDP adoption), (2) reflecting on final driver results, and (3) reflecting on final inhibitor results. All interviews were recorded, transcribed, and analyzed using the guidelines of [51]. After these follow-up interviews, we decided to remove one expert from the study as his organization only adopted LCDPs for data analytics and not for application development.

IV. RESULTS

A. STUDY PARTICIPANTS

In the result section, we first outline the study participants and, second, answer the RQs for drivers and inhibitors.

The response rate for the initial participation call was ~52%. In total, 15 experts participated in the interviews. Of these, nine are consultants, three are line managers, and three are sales executives. After the interviews, two additional experts were identified through one interviewee. Hence, as shown in table 1, 17 experts were invited to participate in the subsequent Delphi study, with nine consultants, five line managers, and three sales executives for LCDPs. Overall the panel size can be considered sufficient for a Delphi study, as it is between seven and 30 [16]. Moreover, we conducted follow-up interviews with four experts (two consultants and two line managers). Appendix B shows biographical information about the experts, the expert group, and their participation in the different rounds. Throughout the whole study, we ensured anonymity for all experts, and each of them approved publishing the results in a paper.

As shown in table 1, the participation dropped from 14 participants in round one to eight in round three. This reduction in participation through different rounds is typical for Delphi studies [53]. Despite the decrease in participants, we were still above the minimum requirement of seven participants [57]. We decided to stop the Delphi ranking after the third round, as we had reached one stop-criteria (i.e., three Delphi rounds [16]). Due to the declining number of participants, we saw it as unlikely that we could motivate the experts for a fourth round. If not otherwise mentioned, we refer to the results for the third Delphi round in the following sections.

TABLE 2. Overview of drivers for LCDP adoption.

Category	Driver	Description
People	Reduction of dependence on internal and external IT developers	An LCDP is adopted to reduce dependence on internal and external IT developers, as citizen developers can develop applications independently.
	Reduction of entry barriers for application development	An LCDP is adopted because an organization wants to reduce the entry barriers for application development (e.g., empowerment of citizen developers).
	Reduction of required knowledge for application development	An LCDP is adopted because less knowledge is required to develop applications (e.g., reduced training effort to develop applications).
Structure	Improved business process efficiency	An LCDP is adopted to improve internal business processes' efficiency (e.g., replace multiple error-prone Excel sheets with one transparent process developed in an LCDP; create applications that would not be developed without the LCDP).
Task	Reduction of Shadow IT development	An LCDP is adopted to reduce Shadow IT development, as citizen developers have a location to develop applications in the LCDP, and all applications are registered on the LCDP.
	Improved efficiency of software development	An LCDP is adopted to improve the efficiency of the software development process (e.g., increase productivity, reduce cost, react quicker to market demands).
	Improved working mode in software development	An LCDP is adopted to improve the working mode of the software development process (e.g., agile working in cross-functional teams).
	Improved effectiveness of software development	An LCDP is adopted to improve the effectiveness of the software development process (e.g., higher-quality applications).
Technology	Transparency of pricing model	An LCDP is adopted to ease the organization's application pricing model.
	Part of existing licenses	An LCDP is adopted because it is already part of existing license agreements an organization uses (e.g., PowerApps are part of most of the Microsoft enterprise agreements).
	Good integration with existing systems	An LCDP is adopted because existing systems integrate well with the LCDP (e.g., PowerApps is used because other Microsoft products are already extensively used).
	Creation of add-ons for off-the-shelf applications	An LCDP is adopted to reduce internal customization needs of off-the-shelf applications by creating add-ons with LCDPs.

We also have a breakdown of consultants and line managers. However, we do not further elaborate on the sales executives group due to only having one participant in the third Delphi round.

B. DRIVERS

1) IDENTIFICATION OF DRIVERS

Our first research question focuses on empirically identifying drivers for LCDP adoption. With our semi-structured interviews, we could identify 12 drivers for adopting LCDPs. We categorized the drivers into four categories: *people*, *structure*, *task*, and *technology*. In the empirical interview data, we did not find any drivers from the *environment* category.

The categories are defined in section II, and the categorization is based on the expert interviews and the work of [32].

Table 2 provides an overview of these drivers, the description, and the categorization. From the interviews, we identified three drivers from the people category, one from the structure category, four from the task category, and four from the technology category. Appendix C provides a list of consolidated expert statements on these drivers.

2) RELATIVE IMPORTANCE OF DRIVERS

The second research question addresses the relative importance of the drivers we identified through the semi-structured interviews. Fig. 2 presents the results and shows the

Kendall's W	Final rank	Driver	Best/Worst score ¹	Mean rank ²	Top-half rank in percent ³	Trend ⁴
$W_1=0.35$	1 [1 / 1]	Improved efficiency of software development	22 [28 / 29]	2.6 [2.4 / 2.8]	86 [89 / 70]	~
$W_2=0.40$	2 [3 / 3]	Reduction of entry barriers for application development	12 [6 / 7]	4.1 [5.8 / 5.4]	86 [44 / 50]	↑
$W_3=0.50$	3 [6 / 6]	Reduction of required knowledge for application development	8 [2 / 5]	5.2 [5.9 / 5.8]	57 [56 / 50]	↑
	4 [2 / 2]	Improved business process efficiency	5 [17 / 11]	5.2 [4.1 / 4.9]	71 [67 / 70]	↓
	5 [8 / 7]	Good integration with existing systems	5 [-1 / 1]	5.5 [6.6 / 6.1]	43 [44 / 40]	↑
	6 [4 / 5]	Reduction of Shadow IT development	2 [4 / 7]	5.9 [5.8 / 5.8]	43 [56 / 50]	↓
	7 [5 / 8]	Improved effectiveness of software development	7 [5 / -1]	5.9 [5.8 / 7]	43 [44 / 30]	↓
	8 [7 / 4]	Improved working mode in software development	3 [4 / 10]	6.2 [6 / 5.4]	29 [44 / 70]	↓
	9 [10 / 10]	Creation of add-ons for off-the-shelf applications	-2 [-7 / -5]	7.4 [7.9 / 7.3]	14 [11 / 30]	↑
	10 [9 / 9]	Reduction of dependence on internal and external IT developers	-7 [-4 / -3]	8.3 [6.8 / 7.2]	14 [56 / 10]	↓
	11 [11 / 12]	Transparency of pricing model	-22 [-20 / -38]	10.1 [9.7 / 10.9]	14 [11 / 10]	~
	12 [12 / 11]	Part of existing licenses	-33 [-34 / -23]	11.7 [11 / 9.7]	0 [11 / 20]	~

1. Calculated #selected most important - # selected least important - data for this can be found in the Appendix | 2. Average ranking of driver for all experts | 3. Percentage of experts having ranked driver within their "top half" (i.e., Top 6) | 4. Trend is positive (negative) if final rank improves (worsens) between round 2 and 3. Trend is neutral (-) if final rank is constant between round 2 and 3.

FIGURE 2. Ranking of drivers for LCDP adoption from all responses.

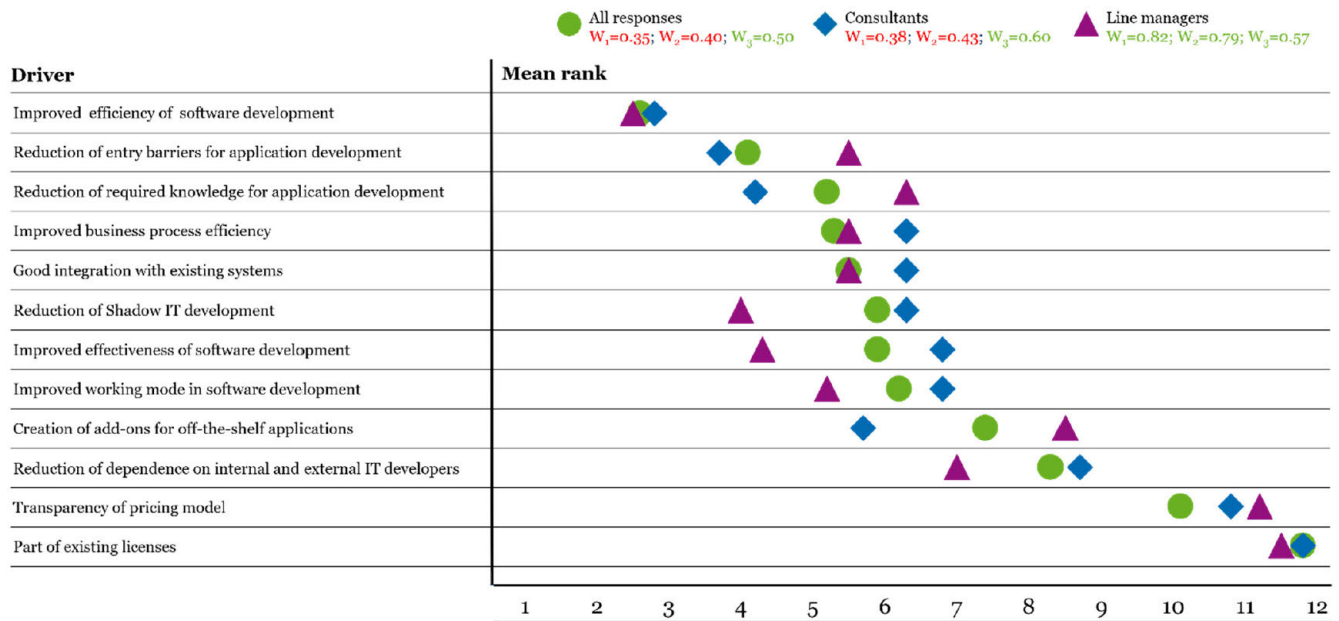


FIGURE 3. Driver breakdown for all responses, consultants, and line managers.

following essential information to report Delphi study results, as outlined by [17] and [18]: Kendall's W (measurement of group consensus), the final rank for each driver (derived from mean rank), the driver, the best/worst score (#most important - #least important), the mean rank (average of

final rankings for each driver from all experts), top-half rank (percentage of experts who ranked the driver in the top half), and the trend. The final rank was determined by the mean rank of each driver, as outlined by [17]. Appendix D provides an overview of how often a driver was selected as most important

and least important for all rounds. For columns with three values, the first value indicates the results of the third round, the second value of the second round, and the third value the results from the first round. When analyzing the quality of the consensus (Kendall's W) for all responses of the third round, we can see that it is 0.50, which indicates a moderate consensus [16]. In their review of the rigor of Delphi studies, [16] found that most Delphi studies (67%) have a Kendall's W between 0.50-0.69.

Interestingly, through all rounds, the top driver (*improved efficiency of software development*) and the bottom two drivers (*transparency of pricing model* and *part of existing licenses*) are similar. Drivers on ranks nine (*creation of add-ons for off-the-shelf-applications*) and 10 (*reduction of dependence on internal and external IT developers*) are also consistently ranked as less important by the experts (i.e., the two drivers change their rank only in the final round). For drivers on the ranks three to eight, the mean rank is relatively close (between 5.2-6.2 in round three), and the ranking of these drivers changes significantly between the different rounds.

As outlined in the methodology section, three different groups of experts participated in the study: consultants, line managers, and sales executives of LCDPs. Fig. 3 compares the ranking of all responses, consultants, and line managers in round three. Due to only having one expert from the sales executives' group participating in the final round, we did not show the breakdown of this group in Fig. 3.

When comparing Kendall's W of all responses with the group consultants and line managers, it is apparent that Kendall's W is higher for the two groups than for all responses. This higher consensus within the groups might result from a higher homogeneity of answers within the groups [58]. As outlined above, consultants and line managers strongly agree on the most and two least important drivers.

Moreover, in Fig. 3, the proximity of the mean ranks (drivers on rank three to eight) for all responses can be seen. For some drivers, the mean rank differs significantly between consultants and line managers. The top three with the most significant difference in mean rank are the *creation of add-ons for off-the-shelf applications* (difference in mean rank: 2.8), *improved effectiveness of software development* (difference in mean rank: 2.5), and *reduction of Shadow IT development* (difference in mean rank: 2.3).

When analyzing the mean rank of the drivers in the four categories of people, structure, task, and technology (cf. table 2), the experts rank technology drivers (mean rank: 8.7) as significantly less important than people (mean rank: 5.9), structure (mean rank 5.2), or task drivers (mean rank 5.1). The overall trend is the same when splitting these results into two groups. Consultants see structure drivers as most important (mean rank 4.0 vs. 6.5 for line managers), whereas line managers see task drivers as most important (mean rank 4.0 vs. 5.7 for consultants). Experts from both groups agree that technology drivers are the least important, with a mean rank

of 8.7 for consultants and 9.2 for line managers. An overview of the mean ranks for all categories and all groups can be found in Appendix E.

C. INHIBITORS

1) IDENTIFICATION OF INHIBITORS

With our semi-structured interviews, we identified 19 inhibitors for LCDP adoption, as shown in table 3. We categorized the inhibitors into five categories: *environment*, *people*, *structure*, *task*, and *technology* based on the expert interviews and the work of [32]. The categories are defined in section II. We identified one inhibitor from the environment category, four from the people category, three from the structure category, one from the task category, and 10 from the technology category. Appendix F provides the consolidated expert statements on these inhibitors.

2) RELATIVE IMPORTANCE OF INHIBITORS

RQ2 addresses the importance of the identified inhibitors for LCDP adoption. Fig. 4 shows Kendall's W, the final rank for all rounds, the inhibitors, the best/worst score, the mean rank, the top-half rank, and the trend. Appendix G gives an overview of how often each inhibitor was selected as most and least important in all rounds. The final rank was determined by the mean rank of each inhibitor, as outlined by [17]. If a column has three values, the first value is the result of round three, the second value is the result of round two, and the third value is the result of the first round. For all responses, Kendall's W is 0.44, which indicates a weaker consensus amongst the experts [16].

Interestingly, the experts agree on the most important inhibitor (*lack of LCD culture and reluctance to change*) and the least important inhibitor (*lack of documentation*), which are ranked consistently throughout rounds two and three. Moreover, experts also see the inhibitors on rank 17 (*lack of scalability*) and rank 18 (*lack of use cases for LCDPs*) as less important through the three ranking rounds. All experts have consistently ranked those three in the bottom half. We also see a high consensus of the experts' rankings for those four inhibitors (the top one and bottom three).

However, the consensus and consistency of the ranking for inhibitors on the ranks two to 16 are lower. For instance, the inhibitor *fear of lock-in to an LCDP vendor* was ranked 12th in the first round, third in the second round, and sixth in the third round. The mean ranks of the inhibitors on ranks two to 16 are close (i.e., 15 inhibitors with an average difference in mean rank of only ~ 0.44). Despite these minimal differences in mean rank, five inhibitors are ranked in the top half (i.e., in the top nine) in all rounds: *lack of LCD culture and reluctance to change*, *fear of security, compliance, and privacy risk*, *lack of governance*, *lack of LCDP developers*, and *too complex development for citizen developers*.

Moreover, six inhibitors are ranked in the bottom half (i.e., bottom nine) in all rounds: *limited functionality of LCDPs*, *limited usability of the to-be-developed applications*,

TABLE 3. Overview of inhibitors for LCDP adoption.

Category	Inhibitor	Description
Environment	Lack of regulatory approval	An LCDP adoption is inhibited due to missing regulatory approval to use LCDPs (e.g., LCDP are not approved for use in highly regulated industries).
People	Fear of island application landscape	An LCDP adoption is inhibited due to the fear of creating island application solutions (e.g., applications with decentral data storage).
	Too complex development for citizen developers	An LCDP adoption is inhibited as the development of applications is seen as too complex for citizen developers.
	Lack of LCD culture and reluctance to change	An LCDP adoption is inhibited due to a lack of a low code development culture and a reluctance to change behaviors (e.g., IT department does see itself as too superior to develop applications in LCDPs).
	Bad previous experiences with similar tools	An LCDP adoption is inhibited due to bad experiences with similar tools which do not deliver their promise (e.g., stop of Google App-Maker).
Structure	Lack of LCDP developers	An LCDP adoption is inhibited due to a lack of internal or external LCDP developers.
	Lack of governance	An LCDP adoption is inhibited due to a lack of governance (e.g., unclarities who is responsible IT department or business department).
	Lack of sponsorship	An LCDP adoption is inhibited due to a lack of top-level sponsorship.
Task	Limited usability of the to be developed applications	An LCDP adoption is inhibited due to the limited usability of the to be developed future applications (e.g., the Look & Feel of future applications are not as expected).
Technology	Limited portability to other LCDPs	An LCDP adoption is inhibited due to limited portability to other LCDPs (e.g., blueprints of applications cannot be transferred to other LCDPs).
	Limited integration to third-party systems and data	An LCDP adoption is inhibited due to the limited integration capability of the LCDP with existing third-party systems and data (e.g., the back-end system does not allow the creation of an interface for data exchange).
	Limited functionality of LCDPs	An LCDP adoption is inhibited due to the limited functionality of LCDPs (e.g., LCDPs lack functionalities for testing).
	Fear of security, compliance, and privacy risk	An LCDP adoption is inhibited due to the fear of increased security, compliance, and privacy risks (e.g., the run-time environment of a low code application is in the cloud).
	Lack of platform flexibility and customization	An LCDP adoption is inhibited due to a lack of flexibility and customization when developing applications (e.g., only pre-configured components can be used on the LCDP).
	Lack of scalability	An LCDP adoption is inhibited due to a lack of scalability (e.g., LCDPs are more suitable for smaller applications with few users).
	Lack of documentation	An LCDP adoption is inhibited due to a lack of documentation of the platform and developed applications.
	Difficult estimation of total cost	An LCDP adoption is inhibited due to difficulties in estimating total cost (e.g., simple applications are cheap, but specialized functionalities require additional sourcing).
	Lack of use cases for LCDPs	An LCDP adoption is inhibited due to a lack of use cases for LCDPs.
	Fear of lock-in to an LCDP vendor	An LCDP adoption is inhibited due to the fear of vendor lock-in (e.g., specific investments to create knowledge and work with LCDP).

Kendall's W	Final rank	Inhibitor	Best/Worst score ¹	Mean rank ²	Top-half rank in percent ³	Trend ⁴
$W_1 = 0.23$	1 [1 / 1]	Lack of LCD culture and reluctance to change	29 [31 / 36]	1.8 [4 / 3.6]	100 [80 / 73]	~
$W_2 = 0.34$	2 [4 / 3]	Fear of security, compliance, and privacy risk	10 [11 / 16]	6.1 [7.1 / 6.9]	86 [60 / 36]	↑
$W_3 = 0.44$	3 [6 / 6]	Lack of governance	12 [8 / 5]	6.3 [7.7 / 8.7]	86 [60 / 36]	↑
	4 [5 / 4]	Lack of LCDP developers	8 [9 / 7]	7 [7.5 / 8.6]	71 [70 / 45]	↑
	5 [10 / 5]	Fear of island application landscape	6 [-1 / 4]	7.6 [10 / 8.6]	71 [40 / 36]	↑
	6 [3 / 12]	Fear of lock-in to an LCDP vendor	5 [15 / -2]	7.9 [6.9 / 10.8]	57 [70 / 18]	↓
	7 [2 / 2]	Too complex development for citizen developers	8 [20 / 19]	8.5 [5.6 / 6.2]	57 [80 / 64]	↓
	8 [9 / 14]	Lack of sponsorship	5 [0 / -4]	8.5 [9.9 / 11.1]	57 [50 / 18]	↑
	9 [7 / 11]	Difficult estimation of total cost	5 [8 / -2]	8.9 [8.1 / 10.5]	43 [50 / 27]	↓
	10 [8 / 7]	Lack of platform flexibility and customization	1 [4 / 4]	9.4 [8.8 / 8.7]	43 [30 / 45]	↓
	11 [11 / 9]	Bad previous experiences with similar tools	-1 [-1 / -4]	10.7 [10 / 10.3]	43 [40 / 45]	~
	12 [15 / 16]	Limited functionality of LCDPs	-6 [-10 / -12]	11.5 [12.2 / 12.4]	29 [10 / 27]	↑
	13 [14 / 15]	Limited usability of the to be developed applications	-4 [-6 / -7]	11.7 [12 / 11.2]	0 [20 / 36]	↑
	14 [18 / 10]	Lack of regulatory approval	-5 [-23 / -3]	11.9 [15 / 10.4]	29 [0 / 27]	↑
	15 [13 / 8]	Limited integration to third-party systems and data	-4 [-6 / 2]	12.1 [11.7 / 9.1]	29 [20 / 45]	↓
	16 [12 / 13]	Limited portability to other LCDPs	-10 [-4 / -2]	12.8 [11.1 / 11]	14 [30 / 27]	↓
	17 [16 / 17]	Lack of scalability	-8 [-9 / -11]	13.4 [12.3 / 12.7]	0 [10 / 18]	↓
	18 [17 / 19]	Lack of use cases for LCDPs	-22 [-16 / -24]	15.9 [14.1 / 14.6]	0 [0 / 9]	↓
	19 [19 / 18]	Lack of documentation	-29 [-30 / -22]	17.9 [16.4 / 14.6]	0 [0 / 0]	~



1. Calculated #selected most important - # selected least important - data for this can be found in the Appendix | 2. Average ranking of inhibitor for all experts | 3. Percentage of experts having ranked inhibitor within their "top half" (i.e., Top 9) | 4. Trend is positive  (negative ) if final rank improves (worsens) between round 2 and 3. Trend is neutral (-) if final rank is constant between round 2 and 3.

FIGURE 4. Ranking of inhibitors for LCDP adoption from all responses.

lack of regulatory approval, limited portability to other LCDPs, lack of scalability, lack of use cases for LCDPs, and lack of documentation.

Fig. 5 shows a breakdown of the ranking on the groups consultants and line managers. For consultants and line managers, Kendall's W is 0.57, which indicates a moderate consensus [16]. The consensus for consultants and line managers is significantly higher than for all responses due to the higher homogeneity of these groups. For the consultants and line managers, the most important (i.e., lack of low code development culture and reluctance to change) and least important (i.e., lack of use cases for LCDP and lack of documentation) inhibitors are the same.

On the ranks between the most and least important inhibitors, consultants and line managers see the importance of the inhibitors differently. The greatest difference in mean ranks for consultants and line managers is also significantly

higher than for the drivers. The top three inhibitors with the highest difference in mean rank are a difficult estimation of total cost (difference in mean rank: 8.5), too complex development for citizen developers (difference in mean rank: 6.7), and limited portability to other LCDPs (difference in mean rank: 5.7). Yet, when analyzing the factors where the mean rank of consultants and line managers differ the most, we have the following result: line managers rank factors higher, which inhibit the adoption on a larger scale in an organization (i.e., too complex development for citizen developers, limited functionality of LCDPs, and limited integration to third-party systems and data). Consultants tend to rank those inhibitors higher that are evaluated before the adoption [59] (i.e., difficult estimation of total cost, fear of lock-in to an LCDP vendor, and limited portability to other LCDPs).

When analyzing the mean ranks of the inhibitors for the five categories environment, people, structure, task, and

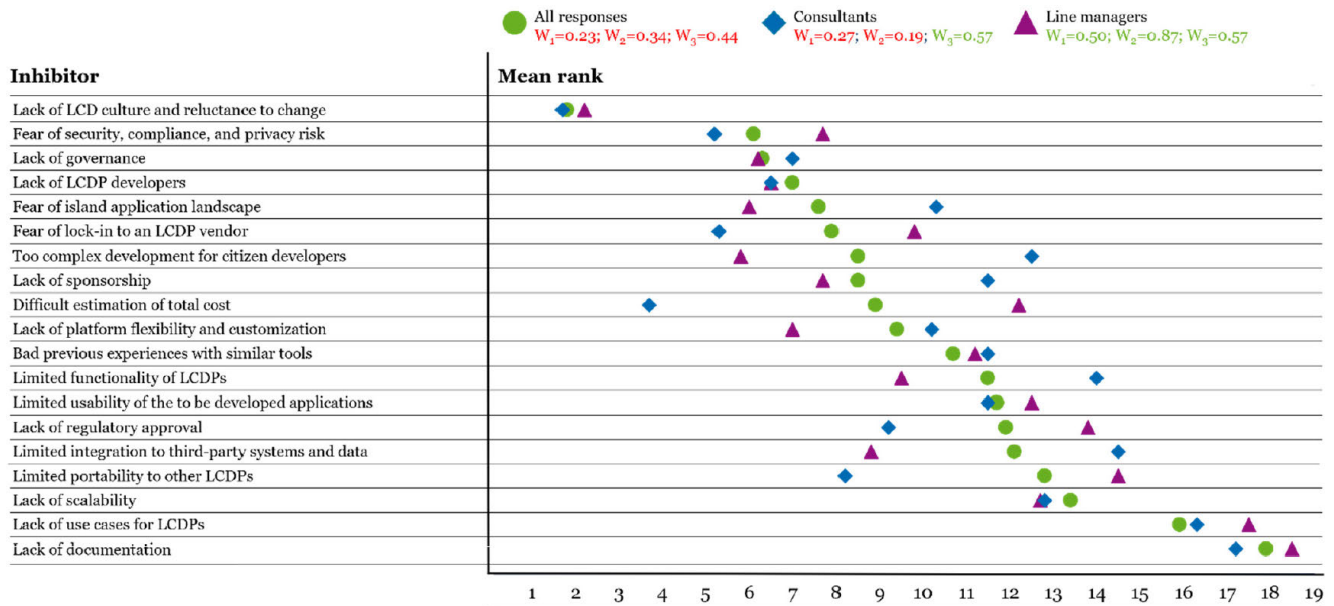


FIGURE 5. Inhibitor breakdown on groups.

technology (cf. table 3), all responses rank environment (mean rank: 11.9), task (mean rank: 11.7), and technology (mean rank: 11.6) inhibitors as significantly less important than structure (mean rank: 7.3) and people (mean rank: 7.2) inhibitors. This result is similar for the groups consultants and line managers. Appendix H provides a detailed visualization of these mean ranks.

V. DISCUSSION

A. OVERARCHING

With our study we empirically researched drivers and inhibitors for LCDP adoption through semi-structured interviews and a Delphi study with 17 experts. Through the semi-structured interviews, we identified 12 drivers and 19 inhibitors for the adoption. In the final round of the Delphi study, we reached a moderate consensus (Kendall's $W = 0.5$) for the drivers and a slightly weaker consensus (Kendall's $W = 0.44$) for inhibitors for all responses. The consensus is higher in the groups consultants (drivers Kendall's $W = 0.6$; inhibitors Kendall's $W = 0.57$) and line managers (drivers and inhibitors Kendall's $W = 0.57$). In the following sections, we will first discuss the context dependency of the ranking results. Then, we will discuss the identified drivers and inhibitors and compare them to the current academic literature.

B. CONTEXT DEPENDENCY

Generally, experts have a strong consensus for the most and least important drivers and inhibitors. However, the consensus is weaker for the factors in between. The adoption's context can explain this result. When we analyzed the collected ranking data, we could find indications for three

different contexts. Due to the little context data collected in the Delphi study, we validated the contexts with follow-up interviews with the experts. The experts confirmed the context dependency (e.g., “*This is probably also due to the context.*” – Expert 14). The three contexts we found, which will be explained in more detail in the next paragraphs, are the *adopted LCDP*, the *expert background*, and the *stage of adoption*.

The *adopted LCDP* impacts the ranking, as license terms, target audiences, capabilities, and functionalities differ significantly between LCDPs. As shown in Appendix I, in the third Delphi round, experts for Mendix and experts for multiple platforms participated. The inhibitors with the biggest difference in mean rank for these two groups are the *bad previous experiences with similar tools*, the *lack of governance*, and the *limited functionality of LCDPs*. When we discussed the results in the follow-up interviews, the Mendix experts stated: “*Limited functionality, lack of scalability, lack of documentation. [...] I don't see this; you can scale, document, [...] there are enough functionalities [...]. Hence, for me, these were no inhibitors for the platform we use.*” – Expert 7. Expert 3 confirmed this notion by stating, “*limited functionality of LCDPs [...] I do not see it as an inhibitor.*” – Expert 3. Hence, the LCDP adopted by these experts seem to be more mature than others, and hence they rank the inhibitor of *limited functionality of LCDPs* as significantly less important. Another notion that was brought up by Expert 7 is license agreements: “*Transparency for the pricing model and part of existing licenses [...] is in our case, through the vendor, [...] we are using Mendix [...] relatively simple. [...] For Microsoft, this is a little bit more difficult [...]. So, I think this highly depends on the vendor, which was chosen or not chosen.*” – Expert 7. We did not have an expert for

Microsoft's LCDP participating in round three, so we do not see this effect in the ranking data. However, as Microsoft's LCDP is part of the Microsoft enterprise license agreement, we agree with the statement. As we showed that the *adopted LCDP* significantly impacts the experts' ranking, we propose to define types of LCDPs with clear characteristics, to increase the comparability of research results.

Moreover, the *expert background* also affects the ranking. We found two different backgrounds: the business department background and the IT department background. If an expert works mainly with the IT department or resides in the IT department, we see the expert as having an IT department background. If an expert works mainly with the business department or resides in the business department, we see the expert as having a business department background. As shown in Appendix J, we had experts with an IT department background and experts with a business department background participating in the third-ranking round. We found that experts with a business department background rank drivers from the task category (i.e., application development-specific drivers) as less important than experts with an IT department background. Expert 11 from a business department answered when asked about the interpretation of the result with: *"From a business perspective, I am not that much interested in how it is getting done. It just needs to be done."* – Expert 11. Expert 7 from an IT department, who ranked three of the four drivers of the task category as the top three, stated: *"I can replicate [my] results. [...] the developer is faster."* – Expert 7. From the result that the expert background has an impact on the ranking, we can derive multiple implications. First, we see that LCDPs are not only adopted by business departments but also by IT departments to develop applications. Second, the type of applications developed differs – experts from the business department see LCDPs as *"a playground for the business departments for quick iterations."* – Expert 15. However, in the IT department LCDPs are also used *"for absolute business-critical apps."* – Expert 7. Nevertheless, if people use LCDPs to develop complex, business-critical applications, they will rank factors from the task category as much more important. Third, when business departments adopt LCDPs, they still rely significantly on the IT department. Expert 7, with an IT department background, summarizes this with: *"People come to us with questions that we should [...] solve or problems that we should solve [...]. And what we are now seeing even more is that we are actively going there and [...] build various centers of excellence so that they are led with a certain governance from us. [...] However, they must organize the people [for the development]."* – Expert 7. To describe such a phenomenon, the concept of Business Managed IT (BMIT) was coined by [60]. BMIT describes a phenomenon where a high degree of IT task responsibility (e.g., development of infrastructure and applications) resides within business departments or in a shared responsibility model with the IT department. Moreover, the business departments manage the IT overtly (i.e., the creation, procurement, and operation of

all software or hardware by the business departments are done in alignment with the IT department) [60]. The overt management of IT by the business departments is required to distinguish BMIT from Shadow IT, where this management is done covertly. Therefore, we can conclude that LCDPs are used as a form of BMIT.

Finally, the *stage of adoption* is a context factor influencing LCDP adoption. PaaS cloud computing adoption [59], [61] distinguishes between the adoption and post-adoption. In the adoption phase, an organization decides to adopt, selects a vendor, and enables the usage within the organization [59], [61]. In the post-adoption, a decision on a project level to adopt PaaS cloud computing is made [59], [61]. As shown in Appendix K, we had experts for the adoption and post-adoption participating in the third ranking round. In the results, we see that experts who focus on the adoption rank the importance of factors higher, which are analyzed when making the adoption decision (e.g., *too complex development for citizen developers, previous bad experience with similar tools, and integration with existing systems*). Post-adoption, project-specific factors (e.g., *lack of (project) governance, difficulty in estimating total costs, and limited portability to other LCDPs*) are significantly more important to post-adoption experts. Hence, the same distinction between adoption and post-adoption, found in PaaS cloud computing adoption, can also be made for LCDPs. This distinction is also reasonable, as LCDPs are specialized PaaS cloud products [10], [21], [22].

C. DRIVERS

Through the semi-structured interviews, we identified 12 drivers for LCDP adoption, which adds six drivers to the ones identified by [12]. These newly identified drivers are *improved business process efficiency, good integration with existing systems, reduction of Shadow IT development, creation of add-ons for off-the-shelf applications, transparency of the pricing model, and part of existing licenses*. Moreover, the experts see the driver *quicker reaction to market demand*, as outlined by [12] a result of *improved efficiency of software development* and not a stand-alone driver. Hence, we decided to follow the experts' advice on this driver and consider the quicker reaction to market demand as part of the improved efficiency.

Through the best/worst scaling of the Delphi study, we could determine the relative importance of the LCDP adoption drivers. The consensus in the final ranking round is moderate (Kendall's $W = 0.5$, which is acceptable as it is in line with most other IS Delphi studies [16]). The higher consensus for the groups consultants ($W = 0.60$) and line managers ($W = 0.57$) is not surprising, as the groups have a higher homogeneity [58]. Moreover, as the number of experts in each group is smaller than for all responses, it is easier to achieve a higher group consensus [16].

We compared the ranking from the experts with the results from the literature review of [12]. We decided to use the

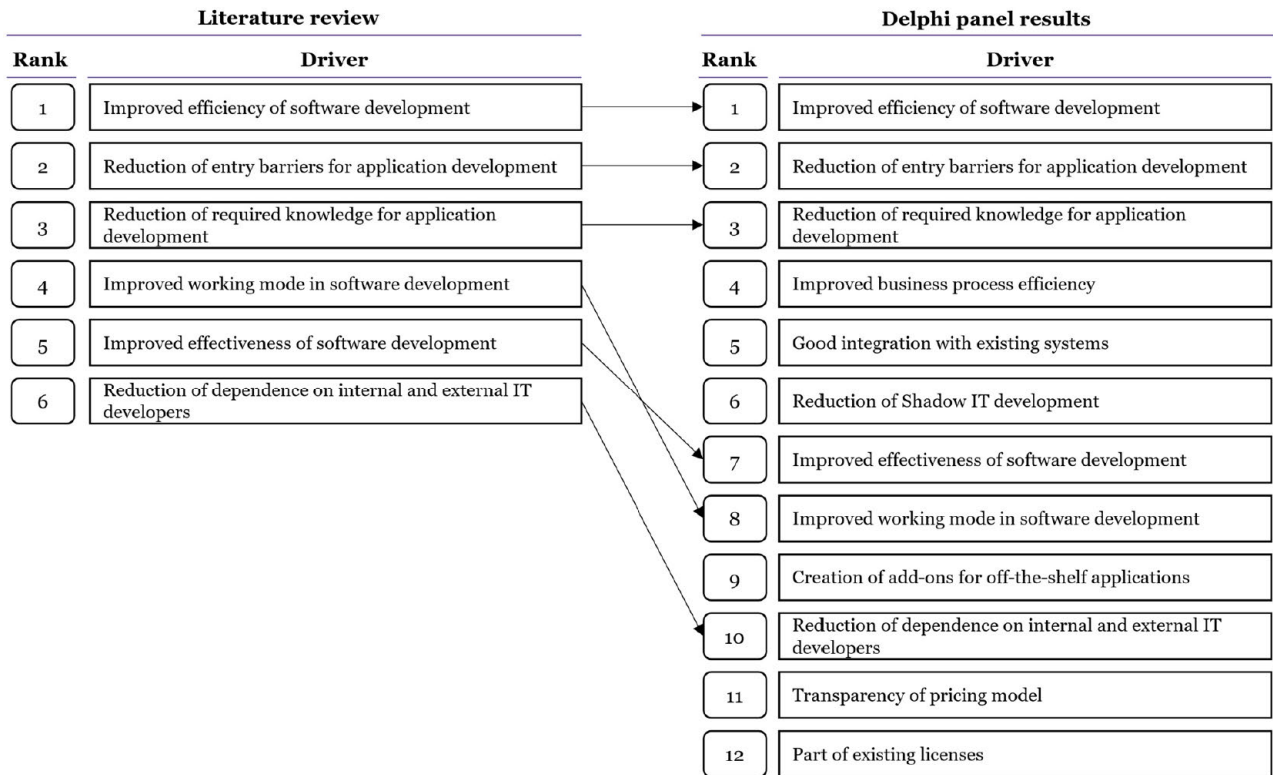


FIGURE 6. Comparison of final driver Delphi results with literature.

review of [12] as it is the most comprehensive review of drivers for LCDP adoption in the academic literature. To determine the literature ranking, we took the findings from [12] and made a simple count analysis (i.e., how often each driver is mentioned in the literature). The results of this comparison can be seen in Fig. 6. Our Delphi study is consistent with the literature for the top three drivers. Yet, for the other drivers, the ranking differs mainly driven through newly identified drivers (e.g., drivers on ranks four to six).

When comparing the ranking of the four driver categories (people, structure, task, and technology), it is apparent that literature focuses on the less important aspects. Current discussions in literature mainly focus on technological aspects of LCDPs [10], [12], whereas the experts rank the importance of these drivers as significantly lower than the drivers of the three other categories. Researching a phenomenon's technology aspects first is common for IS research (similar effects found e.g., [62] for cloud computing or [63] for process mining). We explain that for technological phenomena, first, the technological aspects must be defined, and the phenomena must be conceptualized. After the initial set of technological research questions are answered and the technological aspects of the phenomena are defined, one can research further aspects (e.g., the people aspects of the adoption). If one would first research the people aspects of the adoption and then the technological aspects, there would be a risk in researching the adoption of a concept that is not

fully defined. Moreover, it is not surprising that when doing a literature review on a technology phenomenon (e.g., LCDPs), a significant number of results focuses on the technological aspects. When we showed the results, one expert stated: "I am actually not surprised by this." – Expert 7.

D. INHIBITORS

We identified 19 inhibitors for LCDP adoption with semi-structured interviews. Six have not yet been discussed in academic literature in the context of LCDP adoption. These six are *lack of LCDP developers*, *too complex development for citizen developers*, *lack of sponsorship*, *bad previous experience with similar tools*, *lack of regulatory approval*, and *lack of use cases for LCDP*. In their review on inhibitors for LCDP adoption, [12] outline multiple now-found inhibitors as missing in the academic literature (e.g., sponsorship or regulatory approval). Furthermore, we could empirically confirm the other 13 inhibitors identified by [12].

Through the Delphi study, which followed the semi-structured interviews, we could determine the relative importance of these 19 inhibitors. In the final round, all responses had a Kendall's W of 0.44. This Kendall's W is considered a weaker consensus and below the moderate consensus threshold of 0.50 [16]. However, we could find a moderate consensus for consultants and line managers ($W = 0.57$). A higher homogeneity can explain these results within the groups [58].

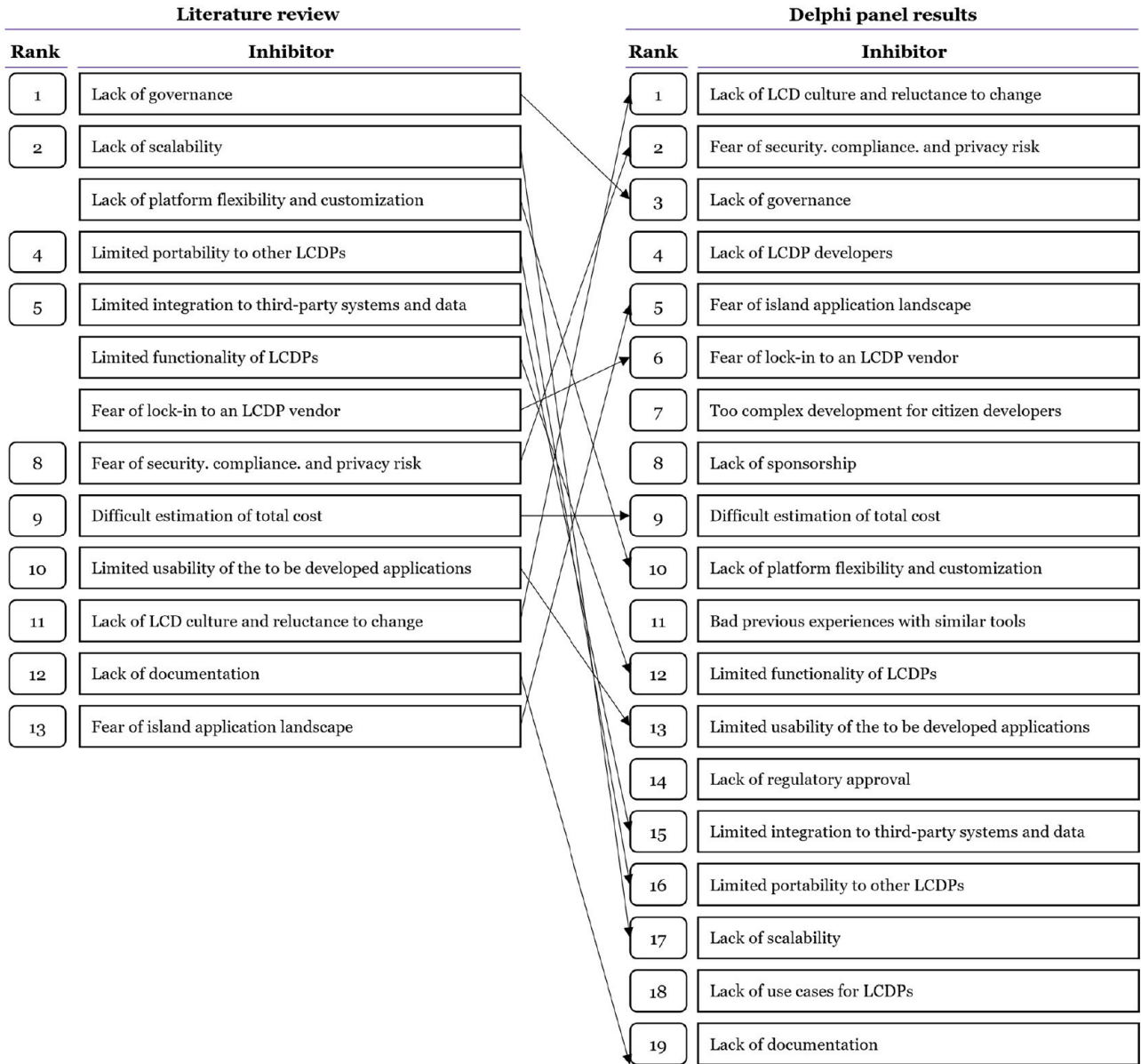


FIGURE 7. Comparison of final inhibitor Delphi results with literature.

We also see a strong consensus for the top one and bottom two inhibitors. For the inhibitors in between, there is disagreement among the experts. As outlined above, we explain this with the context of the adoption situation.

As shown in Fig. 7, the experts' ranking of inhibitors and the importance derived from the literature review of [12] differ significantly. For instance, the most important inhibitor from the Delphi study (*lack of low code development culture*) is only on the 11th rank in the review of [12]. However, in the review [12], the number of occurrences is rather similar (i.e., only one occurrence more or less determines the rank). Only the inhibitor *difficult estimation of cost* is ranked the same by the experts and academic literature. The results show that it is not sufficient to rely on a simple count

analysis to explain certain inhibitors' importance. Moreover, a simple count analysis reveals little about the level of consensus [17].

The academic research on LCDPs mainly focuses on the technology [10]. However, in our Delphi study, the experts rank these inhibitors as less important (mean rank of technology factors: 11.6) compared to the structure (mean rank: 7.3) and people (mean rank: 7.2) inhibitors. The experts explain this result with the relative maturity of the LCDPs from a technology perspective. One expert puts it: "[...] *limited functionality of LCDPs, yes I saw that as less important. [...] I did not really have the limits.*" – Expert 3. Another explanation for this result could be that previously existing technology challenges are now solved

Considering only the following 5 inhibitors for the adoption of LCDPs which is the Most Important and which is the Least Important?

(1 of 19)

Most Important		Least Important
<input type="radio"/>	Lack of platform flexibility and customization <u>Description:</u> LCDP adoption is inhibited due to a lack of flexibility and customization when developing applications (e.g., only pre-configured components can be used on the LCDP).	<input type="radio"/>
<input type="radio"/>	Lack of use cases for LCDP <u>Description:</u> LCDP adoption is inhibited due to a lack of use cases for LCDPs.	<input type="radio"/>
<input type="radio"/>	Fear of security, compliance, and privacy risk <u>Description:</u> LCDP adoption is inhibited due to the fear of increased security, compliance, and privacy risks (e.g., the run-time environment of a low code application is in the cloud).	<input type="radio"/>
<input type="radio"/>	Lack of low code development culture and reluctance to change behaviors <u>Description:</u> LCDP adoption is inhibited due to a lack of a low code development culture and a reluctance to change behaviors (e.g., IT department does see itself as too superior to develop applications in LCDPs).	<input type="radio"/>
<input type="radio"/>	Fear of island application landscape <u>Description:</u> LCDP adoption is inhibited due to the fear of creating island application solutions (e.g., multiple applications with decentral data storage).	<input type="radio"/>

FIGURE 8. Exemplary best/worst scaling questions.

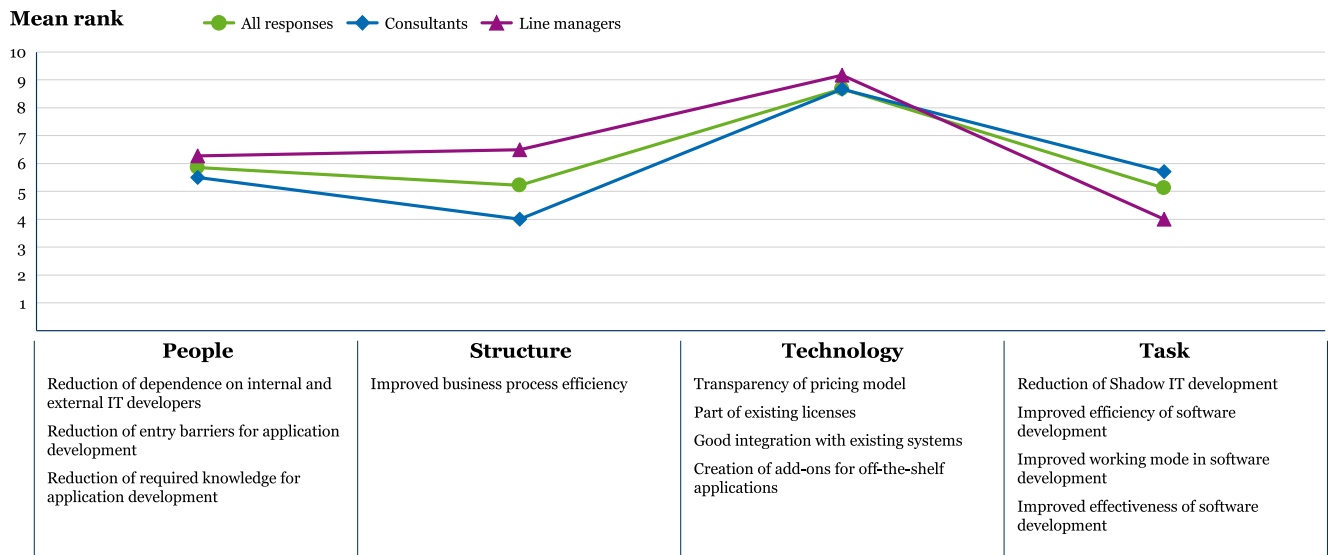


FIGURE 9. Breakdown of mean ranks of drivers on categories.

due to extensive research in this field. Hence, they can no longer be considered important inhibitors as they are now solved.

VI. CONCLUSION

This study aimed to (1) identify drivers and inhibitors for LCDP adoption and (2) determine their importance.

TABLE 4. Overview of experts for this study.

ID	Group	Position	Participation				
			Interviews	Round 1	Round 2	Round 3	Follow-up
1	Sales Executive	Sales executive	Yes	Yes	Yes	Yes	
2	Consulting	Junior partner	Yes		Yes		
3	Consulting	Self-employed LCD expert	Yes	Yes	Yes	Yes	Yes
4	Consulting	Executive at LCD consulting company	Yes			Yes	
5	Consulting	Junior partner	Yes	Yes			
6	Consulting	Junior partner	Yes	Yes			
7	Line Manager	LCDP owner	Yes	Yes	Yes	Yes	Yes
8	Consulting	Senior project manager	Yes ^a	Yes	Yes	Yes	
9	Consulting	Senior project manager	Yes ^a	Yes	Yes		
10	Consulting	Executive at LCD consulting company	Yes	Yes	Yes		
11	Consulting	Senior LCD expert	Yes	Yes	Yes		Yes
12	Sales Executive	Head of sales	Yes				
13	Sales Executive	Head of sales	Yes	Yes	Yes		
14	Line Manager	LCDP owner	Yes	Yes			
15	Line Manager ^c	LCDP owner	Yes	Yes	Yes	Yes	Yes
16	Line Manager	LCDP owner		Yes ^b	Yes	Yes	
17	Line Manager	LCDP owner		Yes ^b	Yes	Yes	

a: Expert 8 and expert 9 were interviewed in one joint interview

b: Expert 16 and 17 joined after the interviews for the Delphi study

c: The results of expert 15 were removed after the final follow-up interviews. This expert's definition of LCDPs is different from this study.

To achieve this goal, we conducted semi-structured interviews and a subsequent best/worst ranking-type Delphi study with 17 experts. The experts were from the three expert groups consultants, line managers, and sales executives of LCDPs.

The study has three main contributions that are beneficial for practitioners and researchers. First, we explicitly address the calls to research LCDPs [10], [12] and their adoption [12] empirically. We have identified 12 drivers and 19 inhibitors for LCDP adoption and added six new drivers and six new inhibitors to the body of knowledge. Furthermore, we can empirically confirm the drivers and

inhibitors discussed in the literature review of [12]. This empirical extension of drivers and inhibitors helps to understand the adoption of LCDPs more substantially. Furthermore, for research to be valuable for practitioners, it needs to be practically useful, and help practitioners make immediate decisions [64]. To be practically useful, the findings must be detailed and specific to the situation (i.e., LCDP adoption). By defining the drivers and inhibitors based on the interviews conducted with the experts and deriving a detailed list of drivers and inhibitors for LCDP adoption, we add substantially to the practical usefulness of our findings.

TABLE 5. Expert statements on the identified drivers.

Driver	Expert statements – selection
Reduction of dependence on internal and external IT developers	"Why they use low code platforms is because organizations [...] find it beneficial to be able to develop workflows and customizations without having developers present. And it's especially advantageous, too, because they often have difficulties actually finding developers and recruiting them." – Expert 6
Reduction of entry barriers for application development	"And that is the reason why they now want to switch to low code because it has the advantage that business people can also be involved in software development [...]" – Expert 1
Reduction of required knowledge for application development	"One thing [...] that also drives you to a low code platform and also to us is the topic of skills. You don't have the latest, the newest skills, the latest technologies, the latest tech stack in every company. [...] And the companies, the employees don't have the skills at all. They don't know the programming languages at all. [...] They don't have the skills to deploy an app like that. And that's what a platform like this takes away from you. " – Expert 12
Improved business process efficiency	"Often it is processes that represent a single source of truth, and it replaces a whole series of updates of Excel sheets [...] and to cover the process fluidly, without error updates." – Expert 1
Reduction of Shadow IT development	"But of course, I have the advantage that I can get rid of all the Shadows IT that exist at the moment" – Expert 4
Improved efficiency of software development	"And that speeds up the app delivery many times over. [...] In my experience, IT projects can be completed up to 10 times faster and with [...] up to 70 percent fewer resources." – Expert 1
Improved working mode in software development	"Because you simply bring business and IT closer together. [...] Because for business in particular [...] they very quickly not only get a feeling for what might be coming, but they also see it. And it's tangible. [...] I mean, you become faster because you shorten the misunderstanding cycles. On the other hand, you really do bring these two parties, business and IT, together." – Expert 8
Improved effectiveness of software development	"So basically, what I'm saying, when you have [...] a low code development platform [...], and you have a business analyst working on it, [...] the business analyst usually has a lot deeper knowledge of the business processes than a typical IT person. And hence you have probably better applications in the end because it's much closer to the real business flow compared to previously when you have to tell it the IT, and then you obviously create a lot of friction." – Expert 6
Transparency of pricing model	"So actually, the costs are relatively manageable. [...] you can use their cloud platform, you use all their resources [...] also for deploying, for creating user stories [...]. So all in one, just for the whole thing. And then the costs are manageable. That is, it's just what you pay Mendix with licensing fees for the entire environment." – Expert 3
Part of existing licenses	"We have large licenses and many services [...], and then, yes, Microsoft does a clever job of integrating these services. If I say I want to edit a form in SharePoint, that's a PowerApp in the background. And then there was already the point where we said, [...] either we stop now and we prevent people from using it [...], or we say and observe that and support it [...]" – Expert 13
Good integration with existing systems	"The thing is open and has interfaces with standard technologies or best-in-class technologies." – Expert 12
Creation of add-ons for off-the-shelf applications	"Which, for me, speaks very strongly for this low code. It's not about replacing off-the-shelf application; it's about actually making the off-the-shelf application more valuable." – Expert 4

TABLE 6. Most important and least important selection of drivers.

Driver	Round 3			Round 2			Round 1		
	#Most important	#Least important	Best/worst score	#Most important	#Least important	Best/worst score	#Most important	#Least important	Best/worst score
Improved efficiency of software development	22	0	22	28	0	28	29	0	29
Reduction of entry barriers for application development	12	0	12	9	-3	6	8	-1	7
Reduction of required knowledge for application development	8	0	8	7	-5	2	12	-7	5
Improved business process efficiency	7	-2	5	17	0	17	11	0	11
Good integration with existing systems	6	-1	5	4	-5	-1	5	-4	1
Reduction of Shadow IT development	8	-6	2	8	-4	4	11	-4	7
Improved effectiveness of software development	9	-2	7	8	-3	5	7	-8	-1
Improved working mode in software development	6	-3	3	8	-4	4	16	-6	10
Creation of addons for off-the-shelf applications	2	-4	-2	2	-9	-7	4	-9	-5
Reduction of dependence on internal and external IT developers.	2	-9	-7	6	-10	-4	5	-8	-3
Transparency of pricing model	1	-23	-22	5	-25	-20	1	-39	-38
Part of existing licenses	0	-33	-33	1	-35	-34	2	-25	-23

Second, we provide an overview of the importance of the drivers and inhibitors for LCDP adoption. In this vein, we could also show that it is not sufficient to simply rely on a count analysis to determine the importance. However, our Delphi ranking allows practitioners and researchers to steer their attention to the most influential aspects of the adoption and save resources. We empirically show that the experts agree on the most important and least important drivers and inhibitors. However, for the drivers and inhibitors between the most and least important, we show that it is context-dependent if a driver or inhibitor is considered important. Moreover, consultants and line managers differ in their rankings, especially for the context-dependent drivers and inhibitors. From a

practitioner’s perspective, this ranking can be used to address the highly ranked inhibitors to facilitate the adoption of LCDPs. For instance, practitioners can derive that they should focus on building an LCD culture (e.g., LCD communities or LCD bootcamps) and implement measures to handle the organization’s reluctance to change (e.g., by showing lighthouse success cases). When we followed the recommendations of [65] for making academic research accessible for practitioners, and discussed our practitioner-oriented report, they highlighted the results’ usefulness due the high level of detail.

Third, we found that the ranking differs between the results of our Delphi study and current academic literature. Recent

TABLE 7. Expert statements on the identified inhibitors.

Inhibitors	Expert statements – selection
Lack of regulatory approval	"For me, especially in Germany, especially in the public sector, the issue is regulatory. That is a big inhibitor." – Expert 8
Fear of island application landscape	"[Company] rolled out [LCDP] in a big way, and now after a year, they're at the point where they realize, "Yeah, so the Citizen Developers don't exist, and the ones that do exist are building some shit." And now we have a huge landscape of hundreds of little applications that are not integrated and don't work together." – Expert 12
Too complex development for citizen developers	"He has an idea of what he wants for an application. But he has far too little understanding of how to build such an application. That is simply not possible today. [...] what these low-code platforms promise, that you build a small application and that the citizen developer builds it, is absolute nonsense." – Expert 12
Lack of LCD culture and reluctance to change	"That's why that's the organizationally more difficult part at the moment, and at the same time, we've created a culture where it's always, "I only have to think, but I never have to implement." So, you don't have the expectation in many areas that you implement yourself." – Expert 7
Bad previous experiences with similar tools	"Well, there were customers who started very early with low code, so the first movers [...]. You can read about Google App Maker was one of the, let's say, first low code platforms [...], which were already quite mature. And that's where a lot of customers got burned because the service was just discontinued." – Expert 11
Lack of LCDP developers	"What you have as a challenge is that there are not so many [LCDP] developers, especially not German-speaking ones either. That simple. And exactly otherwise, they would probably do much more." – Expert 3
Lack of governance	"Yes. The issue of governance that you have to build up. [...] Yes, so what I said, we have 500 apps that have been created because nothing has been registered in the background, which we now have." – Expert 13
Lack of sponsorship	"In other words, the two dimensions of sponsorship and cultural change are totally weak in our company. That would definitely be an inhibitor for us." – Expert 7
Limited usability of the to be developed applications	"That the developed solutions may not quite have the user experience, [...], and I'll call it the quality for the end user, compared to professionally developed software. That can be a disadvantage." – Expert 9
Limited portability to other LCDPs	"You can port it in the sense of simply rebuilding it if it's not too complex. [...]. But I don't know how far the platforms are with this." – Expert 8
Limited integration to third-party systems and data	"But because this [vendor] solution is now not so breathtaking and you use [third-party], you suddenly have this feedback. And you think: Okay, but now I want to get that on [third-party], and I would like to have it mapped somehow to the feedback, and I would like to have it automatically. I can't do that because [vendor] doesn't connect to [third-party]." – Expert 3
Limited functionality of LCDPs	"But that most platforms actually trigger with simple examples, but then as soon as you really want to do something that is in-depth, they often reach their limits." – Expert 13
Fear of security, compliance, and privacy risk	"They've focused so much on it now that you can build apps so quickly and also the customers [...]. We have fun, and the developers; they can generate a lot of applications quickly. [...] This means that so much is being built, but no quality is being assured." – Expert 3
Lack of platform flexibility and customization	"Whereas [Company] as I understand them. They are over-abstracted, they take away too much of your flexibility and don't give you enough ease of use in return." – Expert 2

TABLE 7. (Continued.) Expert statements on the identified inhibitors.

Inhibitors	Expert statements – selection
Lack of scalability	"If you want to do anything additional to that or if you want to be drastically like, for example, scale performance or scale users, then that's probably were like some of the Low-Code platform's kind of hit their limit." – Expert 7
Lack of documentation	"We don't. Maybe we should. [...] That's why [...] why I think the [...] concept actually is better in getting someone to document everything. And document it in such a way that it stands forever. Even if he's gone. I don't know how big a hurdle that will be[...]" – Expert 7
Difficult estimation of total cost	"The price itself may still be an obstacle for many companies because you can look at the websites of the major providers to see what the pricing models are. That is absolutely meaningless. [...] But it is and remains nevertheless expensive in a small scale." – Expert 8
Lack of use cases for LCDP	"It is still for the [...] CIO or CTO to decide, and they are still not sure about it. Yeah. So they don't see so many vital use cases of doing that." – Expert 10
Fear of lock-in to an LCDP vendor	"One is a kind of fear of vendor lock. [...] if you build in a low code platform, at the end of that, you're still tied to [...] that." – Expert 6

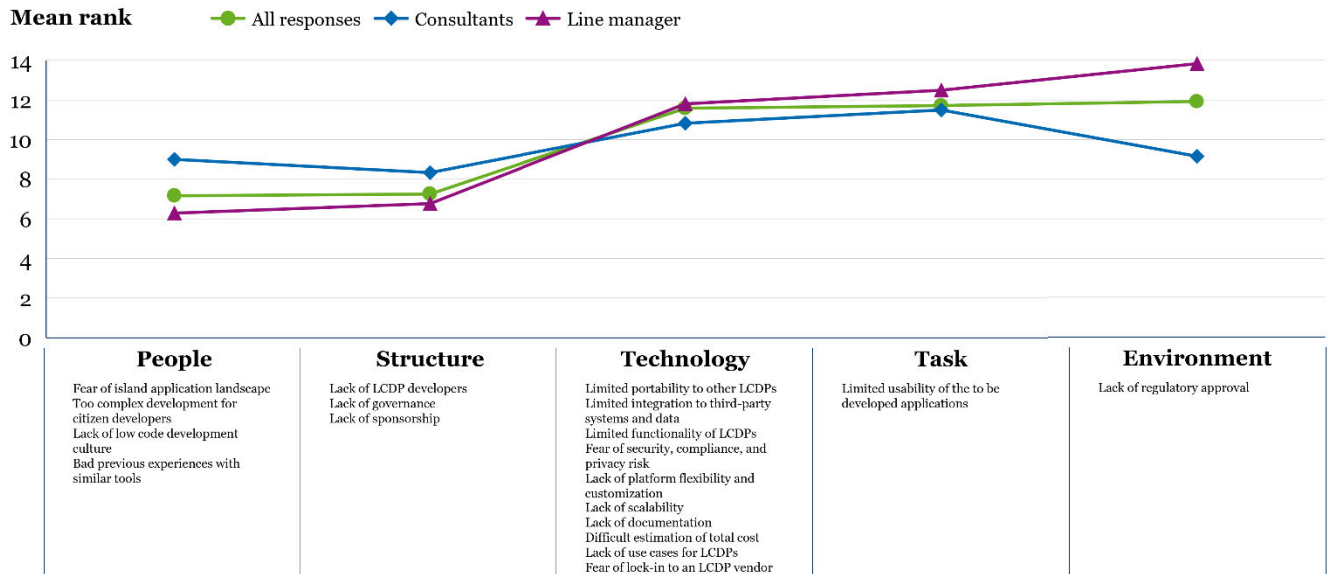


FIGURE 10. Breakdown of mean ranks of inhibitors on categories.

research results are mainly about technological factors ranked as less important by practitioners. Therefore, our results help academia prioritize and focus the research on more important topics from a practitioner’s perspective (i.e., people, structure, and task factors).

As with all academic research projects, this study also has limitations. First, while all experts had significant expertise in adopting LCDPs, most of the experts favored adopting LCDPs. This argument might be especially true for LCDP sales executives. We mitigated this limitation by involving consultants with experience in adoption and non-adoption decisions. Second, the number of participants declined from the interviews to the final ranking round. To mitigate this

limitation, we sent multiple reminders and donated to a charity for participation. However, for the final round, we still had more than seven participants, considered the minimal number of participants for Delphi studies [16]. Third, to keep the results relevant for a business and IT audience, we did not differentiate between specific business or IT functions (e.g., development, maintenance) or an implementation phase. Fourth, as we did not have a guiding theory for this paper and only used the STS theory and TOE model as structuring categories; one might argue that the paper’s theoretical grounding is limited. However, as the focus of this paper is to empirically research drivers and inhibitors for practitioners in detail, we deliberately choose to have a weaker theoretical

TABLE 8. Most important and least important selection of inhibitors.

Inhibitor	Round 3			Round 2			Round 1		
	#Most important	#Least important	Best/worst score	#Most important	#Least important	Best/worst score	#Most important	#Least important	Best/worst score
Lack of LCD culture and reluctance to change	29	0	29	32	-1	31	38	-2	36
Fear of security, compliance, and privacy risk	11	-1	10	14	-3	11	19	-3	16
Lack of governance	14	-2	12	11	-3	8	9	-4	5
Lack of LCDP developers	8	0	8	11	-2	9	12	-5	7
Fear of island application landscape	8	-2	6	6	-7	-1	6	-2	4
Fear of lock-in to an LCDP vendor	5	0	5	20	-5	15	11	-13	-2
Too complex development for citizen developers	13	-5	8	21	-1	20	20	-1	19
Lack of sponsorship	8	-3	5	17	-17	0	14	-18	-4
Difficult estimation of total cost	8	-3	5	11	-3	8	6	-8	-2
Lack of platform flexibility and customization	4	-3	1	6	-2	4	10	-6	4
Bad previous experiences with similar tools	8	-9	-1	8	-9	-1	11	-15	-4
Limited functionality of LCDPs	3	-9	-6	1	-11	-10	3	-15	-12
Limited usability of the to be developed applications	0	-4	-4	6	-12	-6	6	-13	-7
Lack of regulatory approval	3	-8	-5	0	-23	-23	2	-5	-3
Limited integration to third-party systems and data	1	-5	-4	5	-11	-6	9	-7	2
Limited portability to other LCDPs	1	-11	-10	5	-9	-4	11	-13	-2
Lack of scalability	0	-8	-8	3	-12	-9	6	-17	-11
Lack of use cases for LCDPs	0	-22	-22	5	-21	-16	1	-25	-24
Lack of documentation	0	-29	-29	2	-32	-30	0	-22	-22

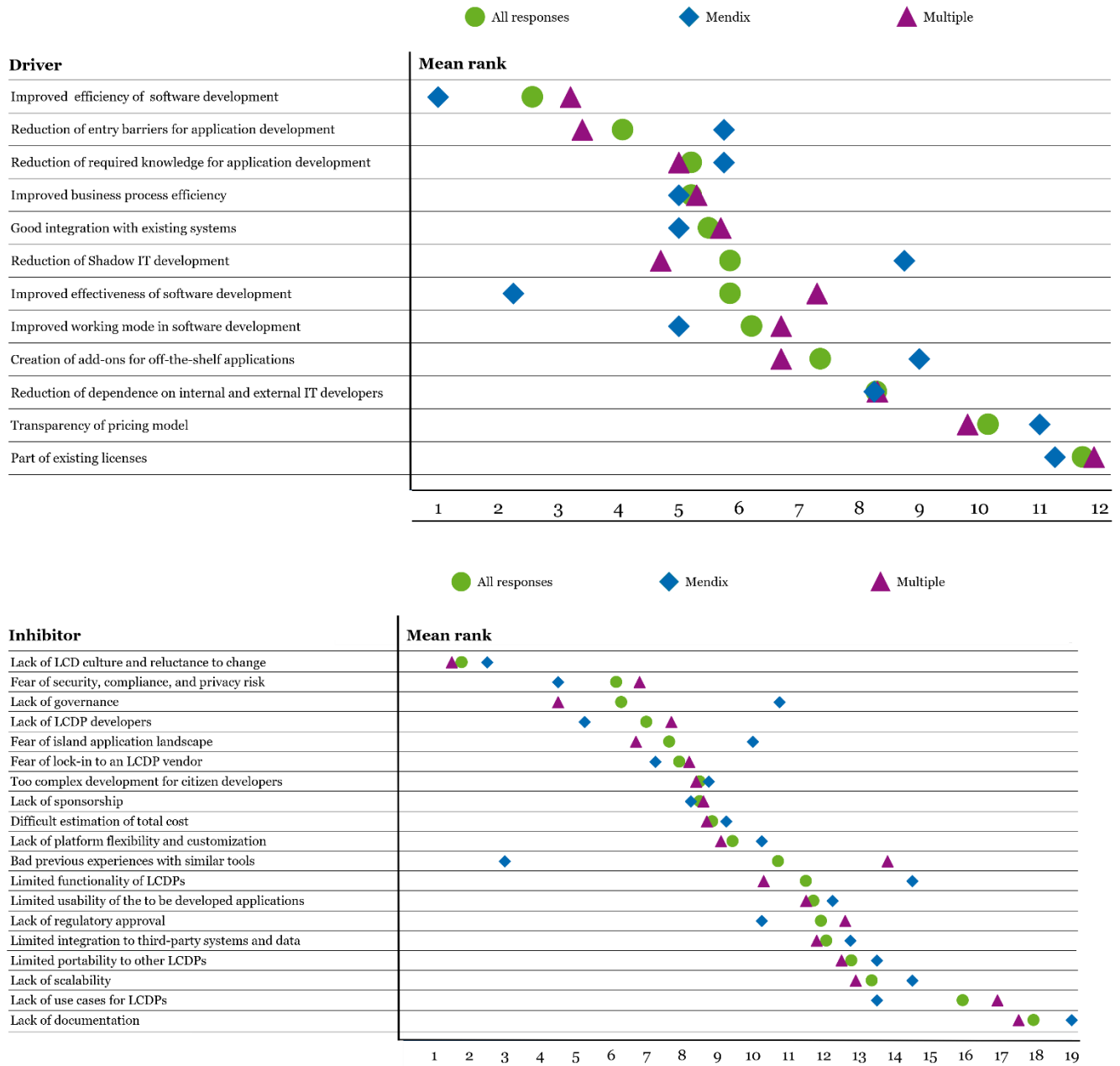


FIGURE 11. Breakdown of drivers and inhibitors on LCDP platform.

grounding. Finally, we found that the ranking is context-dependent. However, the number of experts for each context is small in our data (sometimes only one case). Hence, due to the small number of cases, it is difficult to generalize the context variables. We mitigated this limitation by increasing the richness of data through follow-up interviews with selected experts.

From this study, we can derive multiple areas for future research. The first area is to research the relative impact of the drivers and inhibitors on LCDP adoption or non-adoption. Multiple experts pointed out that no single factor leads to

adoption or non-adoption. As the factors influence each other, one should take a configurational perspective to research this (e.g., [66]). A first step towards this perspective was taken by [22], who built a configurational research model for LCDP adoption and proposed conducting further research using Qualitative Comparative Analysis. Moreover, as there is currently significant technological progress in LCDPs and multiple highly diverse platforms are consolidated under the umbrella term LCDP [6], we propose to create a classification of different LCDPs and distinguish platform types. The definition of LCDP types would benefit the research community,

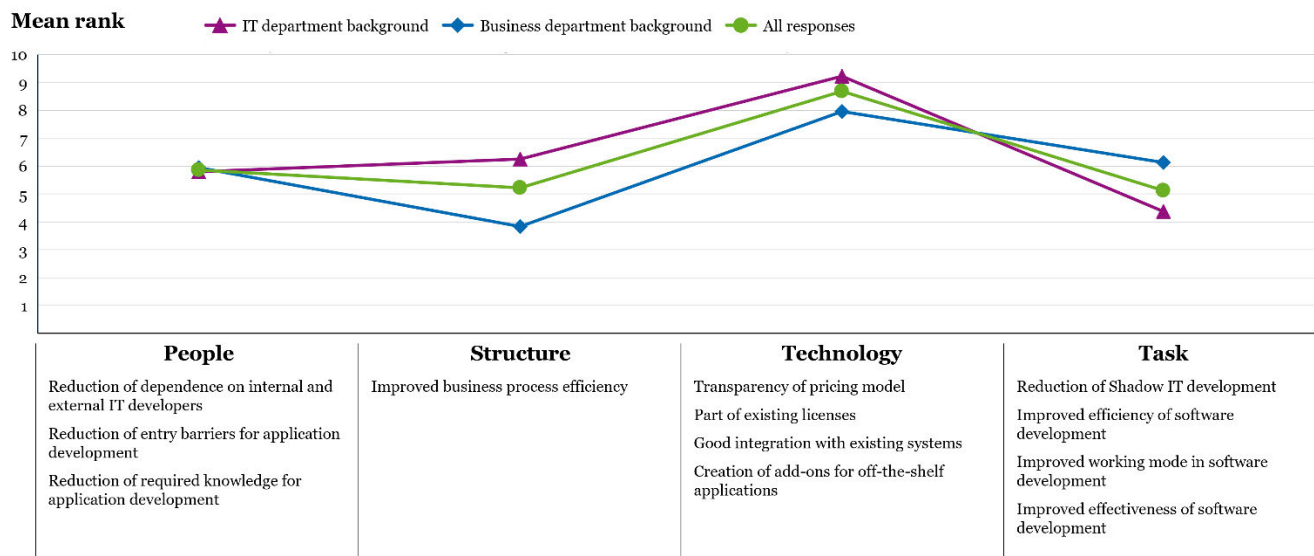


FIGURE 12. Breakdown of drivers on the background.

allowing for more targeted research and better comparability of results. Due to the technological progress, we also propose to re-run the same study in a few years and re-validate our findings. This is justified, as multiple experts argue that they assume to rank the importance differently in a few years due to the fast technological advancement in LCDPs. Finally, as we found strong indications for the effect of contextual factors on LCDP adoption, we propose to research these factors further. This should be done in two ways. First, validate our findings of the three contextual factors with more data. Second, research if there are further contextual factors that we could not identify. This would be beneficial as it would allow for better comparability of research results.

To summarize, this study provides a detailed understanding of what drivers and inhibits the adoption of LCDPs. Further, it shows the importance of the different factors in the adoption decision for LCDPs.

APPENDIX

A. EXEMPLARY BEST/WORST SCALING QUESTION

See Figure 8.

B. OVERVIEW OF THE EXPERTS FOR THIS STUDY

See Table 4.

C. EXPERT STATEMENTS ON THE IDENTIFIED DRIVERS

See Table 5.

D. MOST IMPORTANT AND LEAST IMPORTANT SELECTION OF DRIVERS

See Table 6.

E. BREAKDOWN OF MEAN RANKS OF DRIVERS ON CATEGORIES

See Figure 9.

F. EXPERT STATEMENTS ON THE IDENTIFIED INHIBITORS

See Table 7.

G. MOST IMPORTANT AND LEAST IMPORTANT SELECTION OF INHIBITORS

See Table 8.

H. BREAKDOWN OF MEAN RANKS OF INHIBITORS ON CATEGORIES

See Figure 10.

I. BREAKDOWN OF DRIVERS AND INHIBITORS ON LCDP PLATFORM

See Figure 11.

J. BREAKDOWN OF DRIVERS TO THE BACKGROUND

See Figure 12.

K. BREAKDOWN OF DRIVERS AND INHIBITORS ON ADOPTION AND POST-ADOPTION

See Figure 13.

ACKNOWLEDGMENT

The authors would like to thank Ostbayerische Technische Hochschule (OTH) Regensburg for funding this open-access publication. They would also like to thank all experts who participated in their research project.

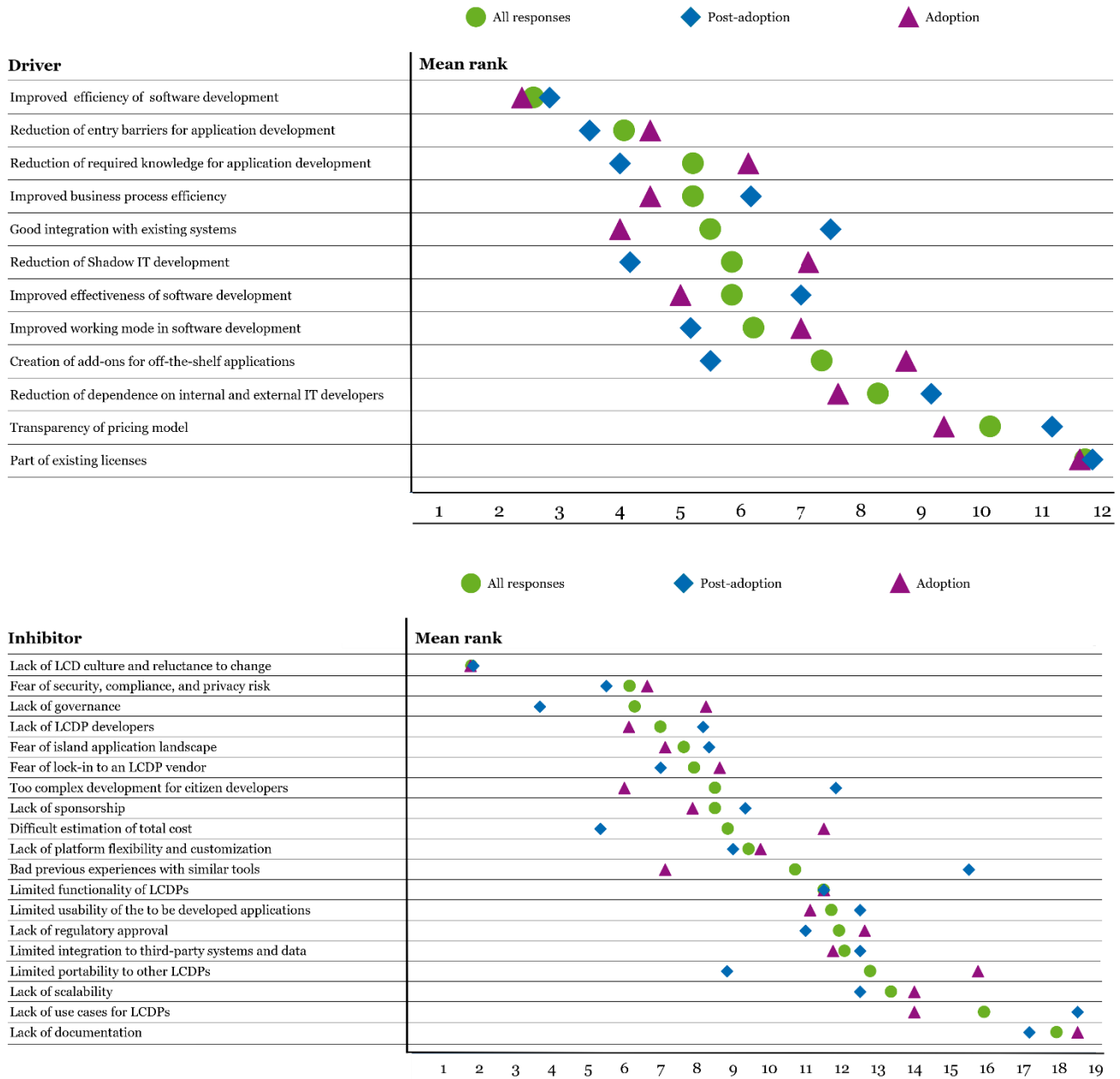


FIGURE 13. Breakdown of drivers and inhibitors on adoption and post-adoption.

REFERENCES

[1] Y. Yoo, O. Henfridsson, and K. Lyytinen, "Research commentary—The new organizing logic of digital innovation: An agenda for information systems research," *Inf. Syst. Res.*, vol. 21, no. 4, pp. 724–735, Dec. 2010, doi: 10.1287/isre.1100.0322.

[2] C. Richardson and J. Rymer. (2014). *New Development Platforms Emerge for Customer-Facing Applications*. Accessed: Feb. 22, 2022. [Online]. Available: <https://www.forrester.com/report/New-Development-Platforms-Emerge-For-CustomerFacing-Applications/RES113411>

[3] E. Sahinaslan, O. Sahinaslan, and M. Sabancıoğlu, "Low-code application platform in meeting increasing software demands quickly: SetXRM," in *Proc. AIP Conf.*, vol. 2334, 2021, Art. no. 070007, doi: 10.1063/5.0042213.

[4] A. C. Bock and U. Frank, "In search of the essence of low-code: An exploratory study of seven development platforms," in *Proc. ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Oct. 2021, pp. 57–66, doi: 10.1109/MODELS-C53483.2021.00016.

[5] M. A. Al Alamin, S. Malakar, G. Uddin, S. Afroz, T. B. Haider, and A. Iqbal, "An empirical study of developer discussions on low-code software development challenges," in *Proc. IEEE/ACM 18th Int. Conf. Mining Softw. Repositories*, May 2021, pp. 46–57.

[6] A. C. Bock and U. Frank, "Low-code platform," *Bus. Inf. Syst. Eng.*, vol. 63, no. 6, pp. 733–740, Dec. 2021, doi: 10.1007/s12599-021-00726-8.

[7] S. S. Bhattacharyya and S. Kumar, "Study of deployment of 'low code no code' applications toward improving digitization of supply chain management," *J. Sci. Technol. Policy Manage.*, vol. 14, no. 2, pp. 271–287, Mar. 2023, doi: 10.1108/JSTPM-06-2021-0084.

[8] M. Bexiga, S. Garbatov, and J. C. Seco, "Closing the gap between designers and developers in a low code ecosystem," in *Proc. 23rd ACM/IEEE Int. Conf. Model Driven Eng. Lang. Systems, Companion Proc.*, Oct. 2020, pp. 1–10, doi: 10.1145/3417990.3420195.

[9] Y.-B. Lin, H. Luo, and C.-C. Liao, "CATalk: An IoT-based interactive art development platform," *IEEE Access*, vol. 10, pp. 127754–127769, 2022, doi: 10.1109/ACCESS.2022.3227093.

- [10] N. Prinz, C. Rentrop, and M. Huber, "Low-code development platforms: A literature review," in *Proc. AMCIS*, 2021, pp. 1–10.
- [11] P. Vincent, "Magic quadrant for enterprise low-code application platforms," Gartner, Stamford, CT, USA, Tech. Rep. G00450466, 2020.
- [12] S. Kass, S. Strahringer, and M. Westner, "Drivers and inhibitors of low code development platform adoption," in *Proc. IEEE 24th Conf. Bus. Informat. (CBI)*, Jun. 2022, pp. 196–205, doi: [10.1109/CBI54897.2022.00028](https://doi.org/10.1109/CBI54897.2022.00028).
- [13] H. A. Alsaadi, D. T. Radain, M. M. Alzahrani, W. F. Alshammari, D. Alahmadi, and B. Fakieh, "Factors that affect the utilization of low-code development platforms: Survey study," *Romanian J. Inf. Technol. Autom. Control*, vol. 31, no. 3, pp. 123–140, Sep. 2021, doi: [10.33436/v31i3y202110](https://doi.org/10.33436/v31i3y202110).
- [14] A. Indamutsa, D. D. Ruscio, and A. Pierantonio, "A low-code development environment to orchestrate model management services," in *Proc. IFIP Int. Conf. Adv. Prod. Manag. Syst.*, 2021, pp. 342–350, doi: [10.1007/978-3-030-85874-2_36](https://doi.org/10.1007/978-3-030-85874-2_36).
- [15] S. Alter, "Work system theory: Overview of core concepts, extensions, and challenges for the future," *J. Assoc. Inf. Syst.*, vol. 14, no. 2, pp. 72–121, Feb. 2013, doi: [10.17705/1jais.00323](https://doi.org/10.17705/1jais.00323).
- [16] G. Paré, A.-F. Cameron, P. Poba-Nzaou, and M. Templier, "A systematic assessment of rigor in information systems ranking-type Delphi studies," *Inf. Manage.*, vol. 50, no. 5, pp. 207–217, Jul. 2013, doi: [10.1016/j.im.2013.03.003](https://doi.org/10.1016/j.im.2013.03.003).
- [17] J. Kobus, M. Westner, S. Strahringer, and D. Strode, "Lean management in IT organizations: A ranking-type Delphi study of implementation success factors," *J. Inf. Technol. Theory Appl.*, vol. 19, no. 3, pp. 82–117, 2018.
- [18] J. Kobus and M. Westner, "Ranking-type Delphi studies in IS research: Step-by-step guide and analytical extension," in *Proc. 9th IADIS Int. Conf. Inf. Syst.*, 2016, pp. 28–38.
- [19] M. Moeini, Y. Rahrovani, and Y. E. Chan, "A review of the practical relevance of IS strategy scholarly research," *J. Strategic Inf. Syst.*, vol. 28, no. 2, pp. 196–217, Jun. 2019, doi: [10.1016/j.jsis.2018.12.003](https://doi.org/10.1016/j.jsis.2018.12.003).
- [20] M. M. Li, C. Peters, K. Eilers, C. Peters, and K. Eilers, "ICT-enabled job crafting: How business unit developers use low-code development platforms to craft jobs," in *Proc. ICIS*, 2022, pp. 1–17.
- [21] J. Rymer, *The Forrester Wave: Low-Code Development Platforms for AD&D*. Cambridge, MA, USA, 2017.
- [22] S. Käss, "Low code development platform adoption: A research model," in *Proc. ACIS*, 2022, pp. 1–8.
- [23] S. Iho, D. Krejci, and S. Missonier, "Supporting knowledge integration with low-code development platforms," in *Proc. ECIS*, 2021, pp. 1–10.
- [24] J. Cabot, "Positioning of the low-code movement within the field of model-driven engineering," in *Proc. 23rd ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst., Companion*, Oct. 2020, pp. 1–3.
- [25] C. Di Sipio, D. Di Ruscio, and P. T. Nguyen, "Democratizing the development of recommender systems by means of low-code platforms," in *Proc. 23rd ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst., Companion*, Oct. 2020, pp. 1–9.
- [26] R. L. Totterdale, "Case study: The utilization of low-code development technology to support research data collection," *Issues Inf. Syst.*, vol. 19, no. 2, pp. 132–139, 2018, doi: [10.48009/2_iis_2018_132-139](https://doi.org/10.48009/2_iis_2018_132-139).
- [27] A. Sahay, D. Di Ruscio, and A. Pierantonio, "Understanding the role of model transformation compositions in low-code development platforms," in *Proc. 23rd ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst., Companion*, Oct. 2020, pp. 1–5, doi: [10.1145/3417990.3420197](https://doi.org/10.1145/3417990.3420197).
- [28] M. A. A. da Cruz, H. T. L. de Paula, B. P. G. Caputo, S. B. Mafra, P. Lorenz, and J. J. P. C. Rodrigues, "OLP: A RESTful open low-code platform," *Futur. Internet*, vol. 13, no. 249, pp. 1–14, 2021, doi: [10.3390/fi13100249](https://doi.org/10.3390/fi13100249).
- [29] T. C. Lethbridge, "Low-code is often high-code, so we must design low-code platforms to enable proper software engineering," in *Proc. Int. Symp. Leveraging Appl. Formal Methods*, 2021, pp. 202–212, doi: [10.1007/978-3-030-89159-6_14](https://doi.org/10.1007/978-3-030-89159-6_14).
- [30] A. M. Khasawneh, "Concepts and measurements of innovativeness: The case of information and communication technologies," *Int. J. Arab Culture, Manage. Sustain. Develop.*, vol. 1, no. 1, p. 23, 2008, doi: [10.1504/ijacmsd.2008.020487](https://doi.org/10.1504/ijacmsd.2008.020487).
- [31] F. Wulf, "Platform-as-a-service post-adoption by large-scale organizations: A research model design," in *Proc. ECIS*, 2020, pp. 1–13.
- [32] N. Prinz, M. Huber, C. Riedinger, and C. Rentrop, "Two perspectives of low-code development platform challenges: An exploratory study," in *Proc. PACIS*, Jul. 2022, pp. 1–16.
- [33] S.-G. Lee, S. H. Chae, and K. M. Cho, "Drivers and inhibitors of SaaS adoption in Korea," *Int. J. Inf. Manage.*, vol. 33, no. 3, pp. 429–440, Jun. 2013, doi: [10.1016/j.ijinfomgt.2013.01.006](https://doi.org/10.1016/j.ijinfomgt.2013.01.006).
- [34] R. Sanchis, Ó. García-Perales, F. Fraile, and R. Poler, "Low code as enabler of digital transformation in manufacturing industry," *Appl. Sci.*, vol. 10, no. 1, pp. 1–17, 2020, doi: [10.3390/app10010012](https://doi.org/10.3390/app10010012).
- [35] Y. Luo, P. Liang, C. Wang, M. Shahin, and J. Zhan, "Characteristics and challenges of low-code development," in *Proc. 15th ACM / IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Oct. 2021, pp. 1–11, doi: [10.1145/3475716.3475782](https://doi.org/10.1145/3475716.3475782).
- [36] M. Sumner and T. Ryan, "The impact of CASE: Can it achieve critical success factors?" *J. Syst. Manage.*, vol. 45, no. 6, p. 16, 1994.
- [37] F. Emery and E. L. Trist, "The causal texture of organizational environments," *Hum. Relations*, vol. 18, no. 1, pp. 21–32, 1965.
- [38] R. Depietro, E. Wiarda, and M. Fleischer, "The context for change: Organization, technology and environment," *Process. Technol. Innov.*, vol. 199, pp. 151–175, Jan. 1990.
- [39] P.-F. Hsu, S. Ray, and Y.-Y. Li-Hsieh, "Examining cloud computing adoption intention, pricing mechanism, and deployment model," *Int. J. Inf. Manage.*, vol. 34, no. 4, pp. 474–488, Aug. 2014, doi: [10.1016/j.ijinfomgt.2014.04.006](https://doi.org/10.1016/j.ijinfomgt.2014.04.006).
- [40] L. Dubé and G. Paré, "Rigor in information systems positivist case research: Current practices, trends, and recommendations," *MIS Quart.*, vol. 27, no. 4, pp. 597–635, 2003.
- [41] A. Strasser, "Delphi method variants in information systems research: Taxonomy development and application," *Electron. J. Bus. Res. Methods*, vol. 15, no. 2, pp. 120–133, 2017.
- [42] R. C. Schmidt, "Managing Delphi surveys using nonparametric statistical techniques," *Decis. Sci.*, vol. 28, no. 3, pp. 763–774, Jul. 1997, doi: [10.1111/j.1540-5915.1997.tb01330.x](https://doi.org/10.1111/j.1540-5915.1997.tb01330.x).
- [43] C. Okoli and S. D. Pawlowski, "The Delphi method as a research tool: An example, design considerations and applications," *Inf. Manage.*, vol. 42, no. 1, pp. 15–29, Dec. 2004, doi: [10.1016/j.im.2003.11.002](https://doi.org/10.1016/j.im.2003.11.002).
- [44] R. K. Yin, *Case Study Research: Design and Methods*, 5th ed. London, U.K.: SAGE, 2014, doi: [10.1111/j.1600-0447.1994.tb05843.x](https://doi.org/10.1111/j.1600-0447.1994.tb05843.x).
- [45] C. Voss, N. Tsikriktsis, and M. Frohlich, "Case research in operations management," *Int. J. Oper. Prod. Manage.*, vol. 22, no. 2, pp. 195–219, Feb. 2002, doi: [10.1108/01443570210414329](https://doi.org/10.1108/01443570210414329).
- [46] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Softw. Eng.*, vol. 14, no. 2, pp. 131–164, Apr. 2009, doi: [10.1007/s10664-008-9102-8](https://doi.org/10.1007/s10664-008-9102-8).
- [47] A. Strasser, "Design and evaluation of ranking-type Delphi studies using best-worst-scaling," *Technol. Anal. Strategic Manage.*, vol. 31, no. 4, pp. 492–501, Apr. 2019, doi: [10.1080/09537325.2018.1521956](https://doi.org/10.1080/09537325.2018.1521956).
- [48] S. Cohen and B. Orme, "What's your preference? Asking survey respondents about their preferences creates new scaling decisions," *Mark. Res.*, vol. 16, pp. 32–37, Jan. 2004.
- [49] S. L. Pan and B. Tan, "Demystifying case research: A structured-pragmatic-situational (SPS) approach to conducting case studies," *Inf. Org.*, vol. 21, no. 3, pp. 161–176, Nov. 2011, doi: [10.1016/j.infoandorg.2011.07.001](https://doi.org/10.1016/j.infoandorg.2011.07.001).
- [50] H. Hanif, A. Hanif, A. Ahsan, A. S. Sadiq, S. Mirjalili, and B. Alkazemi, "Identifying critical dimensions for project success in R&D environment using Delphi study and validation techniques," *IEEE Access*, vol. 9, pp. 133594–133610, 2021, doi: [10.1109/ACCESS.2021.3112112](https://doi.org/10.1109/ACCESS.2021.3112112).
- [51] P. Mayring, *Qualitative Inhaltsanalyse: Grundlagen und Techniken*, 13th ed., Weinheim, Germany, 2015.
- [52] J. Louviere, I. Lings, T. Islam, S. Gudergan, and T. Flynn, "An introduction to the application of (case 1) best-worst scaling in marketing research," *Int. J. Res. Marketing*, vol. 30, no. 3, pp. 292–303, Sep. 2013, doi: [10.1016/j.ijresmar.2012.10.002](https://doi.org/10.1016/j.ijresmar.2012.10.002).
- [53] M. Häder, "Designs von Delphi-Befragungen," in *Delphi-Befragungen*. Wiesbaden, Germany: Springer, 2014, pp. 91–169, doi: [10.1007/978-3-658-01928-0_7](https://doi.org/10.1007/978-3-658-01928-0_7).
- [54] C. Toma and I. Picioreanu, "The Delphi technique: Methodological considerations and the need for reporting guidelines in medical journals," *Int. J. Public Heal. Res.*, vol. 4, no. 6, pp. 47–59, 2016.
- [55] C. A. Brown, "The opt-in/opt-out feature in a multi-stage Delphi method study," *Int. J. Social Res. Methodol.*, vol. 10, no. 2, pp. 135–144, Apr. 2007, doi: [10.1080/13645570701334084](https://doi.org/10.1080/13645570701334084).

- [56] L. Nuño, M. Barrios, E. Rojo, J. Gómez-Benito, and G. Guilera, "Validation of the ICF core sets for schizophrenia from the perspective of psychiatrists: An international Delphi study," *J. Psychiatric Res.*, vol. 103, pp. 134–141, Aug. 2018, doi: [10.1016/j.jpsychires.2018.05.012](https://doi.org/10.1016/j.jpsychires.2018.05.012).
- [57] N. Dalkey, O. Helmer, and O. Helmer, "An experimental application of the Delphi method to the use of experts," *Manage. Sci.*, vol. 9, no. 3, pp. 458–467, Apr. 1963, doi: [10.1287/mnsc.9.3.458](https://doi.org/10.1287/mnsc.9.3.458).
- [58] A. Alarabiat and I. Ramos, "The Delphi method in information systems research (2004–2017)," *Electron. J. Bus. Res. Methods*, vol. 17, no. 2, pp. 86–99, Jun. 2019, doi: [10.34190/JBRM.17.2.04](https://doi.org/10.34190/JBRM.17.2.04).
- [59] F. Wulf, M. Westner, and S. Strahringer, "We have a platform, but nobody builds on it: What influences platform-as-a-Service post-adoption?" *Int. J. Inf. Syst. Proj. Manag.*, vol. 10, no. 1, pp. 49–70, 2022, doi: [10.12821/ijispm100103](https://doi.org/10.12821/ijispm100103).
- [60] A. Kopper, "Business-managed IT: A conceptual framework and empirical illustration," in *Proc. ECIS*, 2018, pp. 1–16.
- [61] F. Wulf, M. Westner, M. Schön, S. Strahringer, and C. Loebbecke, "Preparing for a digital future: Cloud strategy at continental AG," in *Proc. ICIS*, 2020, pp. 1–17.
- [62] H. Yang and M. Tate, "A descriptive literature review and classification of cloud computing research," *Commun. Assoc. Inf. Syst.*, vol. 31, no. 1, pp. 35–60, 2012, doi: [10.17705/1cais.03102](https://doi.org/10.17705/1cais.03102).
- [63] W. M. P. van der Aalst and A. J. M. M. Weijters, "Process mining: A research agenda," *Comput. Ind.*, vol. 53, no. 3, pp. 231–244, 2004, doi: [10.1016/j.compind.2003.10.001](https://doi.org/10.1016/j.compind.2003.10.001).
- [64] A. S. Levine, "Why do practitioners want to connect with researchers? Evidence from a field experiment," *PS, Political Sci. Politics*, vol. 53, no. 4, pp. 712–717, Oct. 2020, doi: [10.1017/S1049096520000840](https://doi.org/10.1017/S1049096520000840).
- [65] M. Marabelli and E. Vaast, "Unveiling the relevance of academic research: A practice-based view," *Inf. Org.*, vol. 30, no. 3, Sep. 2020, Art. no. 100314, doi: [10.1016/j.infoandorg.2020.100314](https://doi.org/10.1016/j.infoandorg.2020.100314).
- [66] Y. Park, P. C. Fiss, and O. A. El Sawy, "Theorizing the multiplicity of digital phenomena: The ecology of configurations, causal recipes, and guidelines for applying QCA," *MIS Quart.*, vol. 44, no. 4, pp. 1493–1520, Dec. 2020.



SEBASTIAN KÄSS received the B.Sc. and M.Sc. degrees in industrial engineering and management from the Karlsruhe Institute of Technology, Karlsruhe, Germany. He is currently pursuing the joint Ph.D. degree with Technische Universität Dresden and Ostbayerische Technische Hochschule (OTH) Regensburg, with a research focus on low-code development platforms.

Before starting in academia, he was as a management consultant at the largest strategy consultancy. He is currently writing his Ph.D. thesis.

Mr. Käss already has several conference publications at ACIS and CBI.



SUSANNE STRAHRINGER received the Diploma in business informatics from the Darmstadt University of Technology, Darmstadt, Germany, where she also obtained her Ph.D. degree and completed her habilitation thesis.

She completed her Habilitation thesis. She is currently a Professor in business information systems, especially IS in manufacturing and commerce with Technische Universität Dresden (TU Dresden), Dresden, Germany. Before joining TU Dresden, she held positions with the University of Augsburg and the European Business School. She has published in *Information & Management*, *Journal of Information Technology Theory and Application*, *Information Systems Management*, and *Journal of Information Systems Education*.



MARKUS WESTNER received the Business Administration and Management Diploma degree from the European Business School, Oestrich-Winkel, Germany, and the master's degree in computing from Unitec, Auckland, New Zealand.

He is currently a Professor in IT management with Ostbayerische Technische Hochschule (OTH) Regensburg, Regensburg, Germany. Before he started his academic career, he was a management consultant in a project manager position for one of the world's largest management consultancies. He is the author of several journal articles and conference papers. His work focuses on IT strategy and IT sourcing.

Prof. Westner acts as an Associate Editor of *Information & Management*.

• • •