6-1-2023

A FRAMEWORK TO ASSESS THE SUITABILITY OF LOW-CODE FOR BPM

Kirill Sadovnikov

Robbert Sweijen

Jan Martijn van der Werf

Inge van de Weerd

A FRAMEWORK TO ASSESS THE SUITABILITY OF LOW-CODE FOR BPM

Research Paper

Sadovnikov, Kirill, KPMG, Amstelveen, The Netherlands, Sadovnikov.Kirill@kpmg.nl.

Sweijen, Robbert, KPMG, Amstelveen, The Netherlands, Sweijen.Robbert@kpmg.nl.

van der Werf, Jan Martijn, Utrecht University, The Netherlands, J.M.E.M.vanderWerf@uu.nl.

van de Weerd, Inge, Utrecht University, The Netherlands, I.vandeweerd@uu.nl.

Abstract

Organizations across all industries seek efficiency, digitization, and automation of their business processes in current times. Low-code development platforms (LCDPs) promise time and cost reduction through rapid and easy-to-use application assembly. Even so, many organizations struggle to understand and identify digital solutions that can advance their business processes. Therefore, we propose a conceptual framework for organizations to assess their business process management (BPM) initiative for LCDP suitability. The framework is developed through a study of literature, a focus group, and expert interviews, resulting in 18 factors to be assessed by organizations. An evaluation using fictitious use case analyses showed that the model was well-received, especially with regard to its completeness and operationality. To the best of our knowledge, this is the first work studying organizational adoption of low-code for the sake of BPM initiatives.

Keywords: Low-code, Business Process Management (BPM), Model-Driven Engineering (MDE), Digital Transformation

1 Introduction

A widespread increase in digital technology adoption has transformed the demands of businesses' customers and employees (van Veldhoven and Vanthienen, 2022; Konopik et al., 2022). Not only do organizations need to react with new digital technologies, they need to digitize their business processes to be able to respond to changing requirements (Denner et al., 2018). To support large, end-to-end processes in a structured and adaptive way, Business Process Management Systems (BPMSs) are proposed as a suitable solution during a digital transformation (Xu et al., 2018; Brkić et al., 2020). BPMSs allow to orchestrate and automate a process, improving process performance and organizational agility (Ravasan et al., 2014; Dumas et al., 2018; Xu et al., 2018).

A fundamental decision that organizations face when implementing a BPMS in their IT infrastructure is whether to build it in-house or acquire a packaged solution from a vendor (Ravesteyn & Batenburg, 2010). This dilemma, known as the traditional 'build vs. buy' decision (Hung & Low, 2008), has been studied elaborately in literature (Rands, 1993; McManus, 2003). Recently, a new type of solution has emerged: Low-code. It is becoming increasingly popular as it promises to combine the flexibility of building a solution, and the efficiency of buying one (Cicman et al., 2021).

Low-code is a method for assembly of Information Technology (IT) applications by eliminating most hand-coding for developers (Richardson & Rymer, 2014). Low-code development platforms (LCDPs)

embrace low-code and offer a visual-based platform where the user interface, business logic, workflow, and data handling can be constructed rapidly through easy-to-use component assembly and modeling (Richardson & Rymer, 2014; Metrôlho et al., 2019; Sahay et al., 2020; Sanchis et al., 2020). Low-code's origins lie in Model-Driven Engineering (MDE) where models form the core of code generation and system development (Cabot, 2020; Bock & Frank, 2021a).

Low-code is seen as a major facilitator of organizations going through the digital transformation (Sanchis et al., 2020) and it is often applied in process automation solutions (Luo et al., 2020). Yet, organizations struggle to understand and identify the digital solutions that can advance their business processes (Denner et al., 2018). No study to date has provided indicators on the conditions when low-code can be a suitable solution to organizations (Bock & Frank, 2021a). In this paper, we aim to close that gap for LCDPs aimed at BPMS development, hereafter referred to as 'low-code BPM'. We propose a conceptual framework for organizations to assess the suitability of their BPM initiative to be supported by low-code BPM.

In the following section, we further conceptualize our definition of low-code BPM and provide theoretical background. Thereafter, we illustrate how design science is employed to create and evaluate our conceptual framework. In Section 4, the framework's structure and content is presented. The results from the framework evaluation are formulated in Section 6. Lastly, in Section 7, the results of the study are discussed together with indications for future research.

2 Theoretical background

Academic literature is yet to provide a clear conceptualization of an LCDP (Bock & Frank, 2021a). In this section, we formulate our definition of low-code BPM's offering. Subsequently, we elaborate on the benefits, disadvantages and adoption reasons of low-code BPM.

2.1 Conceptualization of low-code BPM

In general, LCDPs are platforms that allow rapid application development through low-code (Vincent et al., 2020), an approach where standardized, high-level functional components can be assembled easily in a visual designer, instead of textual coding (Metrôlho et al., 2019; Sahay et al., 2020; Sanchis et al., 2020). Generally, we found that LCDPs consist of three main functionalities: (1) they allow the modeling of (data) system structures and business processes, (2) they provide capabilities to design custom graphical user interfaces (GUIs), and (3) they offer flexible integration with external systems (Sahay et al., 2020; Vincent et al., 2020; Bock & Frank 2021a).

To specify further, Frank et al. (2021) classify LCDPs based on their features into four groups: 'basic data management platforms', 'workflow management systems', 'extended GUI- and data-centric IDEs', and an all-encompassing fourth 'multi-use platforms' group. The first group offers features used to develop data management systems and the third focuses on developing web- or mobile applications, both outside of our scope. 'Workflow management systems' are a logical group to specify low-code BPM, as BPMSs were originally known as workflow management systems (Dumas et al., 2018). Although 'multi-use platforms' embody low-code BPM functionality, we do not further analyze these to elicit LCDPs' offerings when used for BPM owing to its all-encompassing scope.

Frank et al. (2021) propose that workflow management systems specialize in workflow automation, through conceptual modeling languages like Business Process Model and Notation (BPMN) or other structures. Moreover, these platforms provide additional support for workflow execution, and integrations with internal and external systems (Bock & Frank, 2021a). The latter can be systems in the organization's IT landscape or AI (Frank et al., 2021), machine learning (Koplowitz, 2017), or other automating services that the platform provides. Lastly, such systems provide functionality to build analytics dashboards to monitor process performance (Waszkowski, 2019).

To conceptualize what low-code BPM platforms can offer, we combine overall LCDP features with the 'workflow management system' specializations to define low-code BPM in this study as: *Low-code BPM supports rapid application development for end-to-end case management, with the workflow*

model and engine at its core, enabling process automation, integration, monitoring and enhancement with intelligent automation technology through easy-to-use component assembly and modeling.

2.2 Background on low-code BPM

Cai et al. (2022) have shown that automating business processes is possible through low-code and that it reduces manual workload and improves IT flexibility. Waszkowski (2019) describes the design of a "BPM low-code platform" that represented similar functionalities as described in our low-code BPM conceptualization. For the rest, literature on the use of low-code for BPM is scarce.

Alternatively, several studies provide evidence of low-code's benefits and disadvantages in general. Low-code can increase software development speed and adaptability in organizations (Sahay et al., 2020; Sanchis et al., 2020; Frank et al., 2021). This is achieved through complexity reduction in low-code (Alsaadi et al., 2021) which, in turn, allows employees with in-depth business knowledge to be involved (Sahay et al., 2020; Iho et al., 2021). In the case of BPM, this allows experienced process owners to develop and adapt the system to better suit the business process needs. Furthermore, cost reduction, increased maintainability, and improved system quality are seen as reasons for adopting low-code in organizations (Alsaadi et al., 2021; Bock & Frank, 2021a; Frank et al., 2021; Luo et al., 2021).

Literature also discusses the shortcomings of low-code. From a technical perspective, LCDPs have struggled with scalability issues, lack of customization on design and layout, and lack of interoperability between LCDPs (Sahay et al., 2020; Luo et al., 2021). Organizational considerations, such as distrust in the technology's abilities, concerns about a low-code vendor 'lock-in', or a steep learning curve have also hampered its adoption (Sahay et al., 2020; Sanchis et al., 2020; Alsaadi et al., 2021).

For organizations, the question arises under what circumstances the cited benefits outweigh the disadvantages as low-code is not suitable for all problems, in all organizations (Frank et al., 2021). Failing to recognize the right digital technologies can, instead, result in a loss of competitiveness for organizations (Konopik et al., 2022). For other digital technologies in BPM, such as robotic process automation (Plattfaut et al., 2022) or process mining (Mamudu et al., 2022), researchers have investigated what factors allow to assess the suitability of an organization's problem to be supported by a certain technology. We, now, study these for low-code BPM.

3 Research method

Our conceptual assessment framework has been constructed following the design science approach, as it "*creates and evaluates IT artifacts intended to solve identified organizational problems*" (Hevner et al., 2004, p. 77). To structure our research approach, we used the design science research methodology (DSRM) by Peffers et al. (2007) visualized in Figure 1.

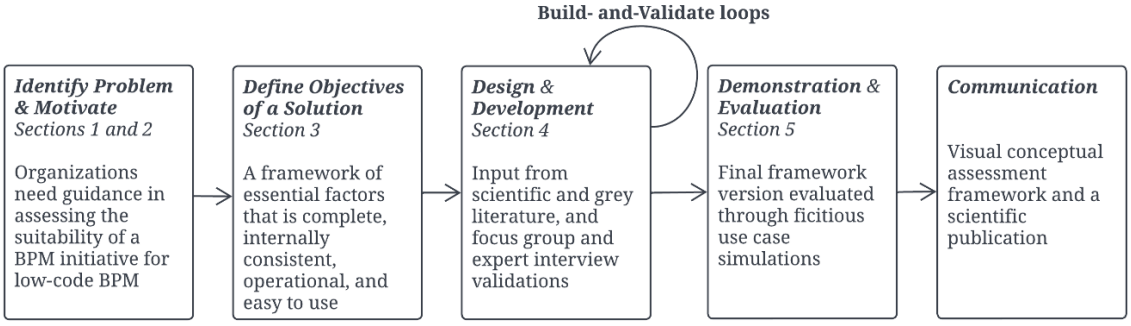

Figure 1. The design science research methodology (Peffers et al., 2007) applied in this study

The first step in the DSRM has been elaborated upon in the previous sections. Organizations need guidance in understanding and evaluating low-code BPM's applicability for their BPM initiative. In the next subsection, we present the solution objectives The conceptual approach to designing and

developing our solution is explained in Section 3.2 while the empirical approach, and evaluation step of the DSRM, are presented in Section 3.3. For more details on the literature, the participants, and how this method was developed, we refer the reader to Sadovnikov (2022).

3.1 Solution objectives

The goal of this study is to design a conceptual framework in the form of an artifact that helps organizations understand and evaluate the suitability of low-code BPM support for their BPM initiative. Peffers et al. (2007) suggest that the criteria for evaluating the artifact should be specified beforehand. To this end, we have integrated criteria from March and Smith (1995) that focus on models and methods, with an emphasis on informational and applicable purpose. Specifically, our framework should provide a complete and consistent picture of all BPM initiative characteristics that differentiate low-code BPM from other solutions, and should enable organizations to assess the applicability of low-code for their business process management initiative through an easy-to-use framework. To clarify our four evaluation criteria - completeness, internal consistency, operationality, and ease of use - we have adopted definitions from Prat et al. (2015).

3.2 Framework construction

In the design and development step, we constructed our framework. The initial version is based on a literature review, whereafter, two build-and-validate loops further refined the framework, as can be seen in Figure 1. The latter is presented in the following subsection.

We have studied and analyzed scientific literature regarding low-code, BPMSs, and MDE, and grey literature on low-code. Literature on BPMS and MDE has been included as low-code BPM aims to construct a BPMS, and low-code has evident origins in MDE. We have included grey literature as many acclaimed market research firms have been analyzing the advancement of low-code in recent years. With their focus on the industry's adoption of low-code, and a scarcity of scientific literature (Sanchis et al., 2020), these reports provide rich data. Only reports from renowned market research firms, large vendors, consultancy firms, or implementation specialists have been selected. The final set contains 41 studies and documents, and has been used to compose a first conceptual assessment framework version.

3.3 Framework validation and evaluation

We have validated our framework in two rounds. The initial framework version has been validated by a focus group, as focus groups can explore a range of different ideas and represent various perspectives among a group of people (Krueger & Casey, 2014). This broadness is deemed useful as a first validation. We used a mini focus group of three people, because our participants had highly specialized knowledge and experiences to discuss. The core question that the focus group had to answer is this study's research question: "How can organizations assess the suitability of their business process management initiative to be supported by a low-code development platform?". Based on the input from the focus group, we developed a refined version of the framework. Three participants have been carefully chosen to have various viewpoints and backgrounds related to low-code development, as presented in Table 1. The size of our focus group is rather small, but in cases where participants have highly specialized knowledge and experiences to discuss this is sufficient (Morgan, 1995).

#	Current Role	Experience with low-code BPM
FG1	Manager in Digital Transformation and Intelligent Automation	Leading and consulting on multiple low-code projects for application development and workflow automation using various low-code development platforms
FG2	Senior consultant in Digital Sourcing and Procurement	Developing and leading the technical implementation for multiple low-code development projects for application development using various low-code development platforms

FG3	Senior manager in Operational Excellence and Digital Transformation	Leading multiple projects on process excellence initiatives including overseeing multiple low-code BPM implementations

Table 1. Focus group participants.

Thereafter, the refined framework has been validated through eight low-code expert interviews. Expert interviews enable more in-depth analyses of topics in which the participant has expertise (Meuser & Nagel, 2009), allowing further refinement of the framework. As criterion sampling, each interviewee has to (*1*) have cooperated on at least one implementation of a low-code solution for BPM, and (*2*) have completed at least one low-code vendor's basic development training. The summary of all participants is given in Table 2. Input from the expert interviews has resulted in the final version of the framework, as presented in this paper.

#	Current role	Experience with low-code BPM
I1	Chief Technical Officer	Setting-up, implementing and overseeing the use of Mendix in their whole organization
I2	Director and IT implementation specialist	Consulting organizations with decision-making and managing IT implementations through OutSystems
I3	Lead Appian consultant	Leading the implementation of Appian throughout a large corporate and consulting small-medium enterprises on Appian
I4	Freelance Business & Digital Transformation expert	Leading the implementation of various LCDPs in various organizations
I5	Appian developer	Building Appian applications for various organizations
I6	Interim Chief Information Officer	Managing and decision-making on IT implementations, including LCDPs, in various organizations
I7	Partnership manager in Intelligent Automation	Connecting organization needs with Intelligent Automation and low-code BPM vendors
I8	Lead in consulting for Intelligent Automation	Setting-up a low-code BPM department and consulting on Intelligent Automation opportunities

Table 2. Expert interview participants.

In the next step of the DSRM, the definitive framework has been demonstrated and evaluated through fictitious use case simulations. An interview and survey at the end of these simulations gauges the criteria from Section 3.1. The entire evaluation method has been described in Section 6.

3.4 Data analysis

We have used thematic analysis as a systematic method, described by Braun and Clarke (2012), to analyze our qualitative dataset, consisting of literature and transcribed validations and evaluations. We used a combination of deductive and inductive coding to derive the framework's contents.

A deductive coding approach uses a pre-existing framework or theory to later find themes of interest in the dataset (Crabtree & Miller, 1999). At first, we have identified two existing models that provided scientifically substantiated dimensions that could be used to categorize our factors. These dimensions are presented in the next section. Thereafter, we analyzed the literature, with the dimensions in mind, and labeled it with codes of interest. We searched and reviewed our codebook for themes, categorized them under the dimensions, and gave the themes a definition and a description, according to the methodology by Braun and Clarke (2012). These themes are eventually included as factors in our conceptual assessment framework.

The analyses of the validations and evaluation followed a similar procedure. However, after deriving the themes during analysis, we revisited the existing factors and their substantiation. As we conducted

and analyzed each method sequentially, contradictions and discrepancies can occur when integrating results (Moran-Ellis et al., 2006). Therefore, throughout our results, we present our rationale on how we reconciled such discrepancies.

4 Low-code BPM assessment framework

The definitive conceptual assessment framework is presented in Figure 2. It includes 18 factors, representing various topics to judge or consider, divided into two factor types. The *key factors* represent the characteristics of an ideal use case for low-code BPM and a use case should 'have' as many of these factors as possible. Organizations should remain realistic and critical when judging their use case against the key factors, and judge what a missing factor implies in their context. The *reflection factors* are topics specifically important when contemplating low-code BPM that an organization should be aware of and consider for themselves. However, contrary to key factors, these can still be 'solved' or mitigated before adoption. Although the framework is not an objective measurement tool, the factors should ascertain organizations in their decision-making when considering low-code BPM.

As mentioned earlier, two existing theories support our framework. Whittle et al. (2017) present a taxonomy of MDE tooling considerations that shape successful MDE adoption and use. Due to low-code's origins in MDE, these themes represent the technological side of our framework. Vom Brocke et al. (2016) provide the BPM context for this framework through the 'morphological box to identify the context of BPM'. By combining the distinct themes in these models, we end up with six dimensions under which each factor in our framework falls.

The dimensions can be found, sometimes rephrased, in the framework as layers separating the key factors. Moreover, key factors have white tick-off boxes so users can structurally go through the framework. We have not included these features for the reflection factors, as these are more intended for a general understanding of important topics when considering low-code BPM.

In the following subsections, we present how findings from literature, input from the focus group and expert interviews are incorporated into the 18 factors. Last, the framework's design is substantiated.

4.1 Goal dimension factors

This dimension describes specific project goals that low-code BPM is especially suitable for.

Process Orchestration (Key) – Integration of the whole end-to-end process is a common problem for organizations in current times (Xu et al., 2018). Our expert interviewees argued that low-code BPM allows this, clearly illustrated by I8: "*The entire case was managed by a process workflow. So the notion of tasks. And I distribute those tasks, I assign them to people, to groups, and they take ownership of that. That is kind of the core of the whole process orchestration, people-in-the-loop. But I have the system driving the business logic, as who should be doing what now, instead of people doing that themselves*". This corresponds with the characteristics of workflow management systems as described by Bock and Frank (2021b) and our low-code BPM conceptualization. A core component to achieve such process orchestration is low-code BPM's integration with the IT environment (Frank et al., 2021). I3 illustrated this with a practical low-code BPM example: "*A know-your-customer process, that often involves multiple departments, having to pull data from multiple sources and bring it all together into one bundle [...] that is a clear process orchestration initiative*". Low-code BPM allows process orchestration to reduce inefficiencies, improve performance, and provide transparency in the process.

Speed (Key) – Low-code development enables organizations to develop products in a short time (Sahay et al., 2020; Alsaadi et al., 2021; Sanchis et al., 2021) and low-code vendors explicitly mention development speed as a key benefit of low-code (OutSystems, 2019). All experts agreed that this is one of the key goals achieved with low-code, also in the case of BPM, as mentioned by I2: "*The competitive edge [of low-code] is found in the speed and the agility it offers […] where you talk about implementation periods of about 1 to 3 months*". I6 had seen it being a decision point: "*At [organization] the reason [for adopting low-code] was that we wanted to develop relatively quickly*" (I6). Low-code BPM can be the answer for organizations that need to develop IT solutions quickly (Sanchis et al., 2020).

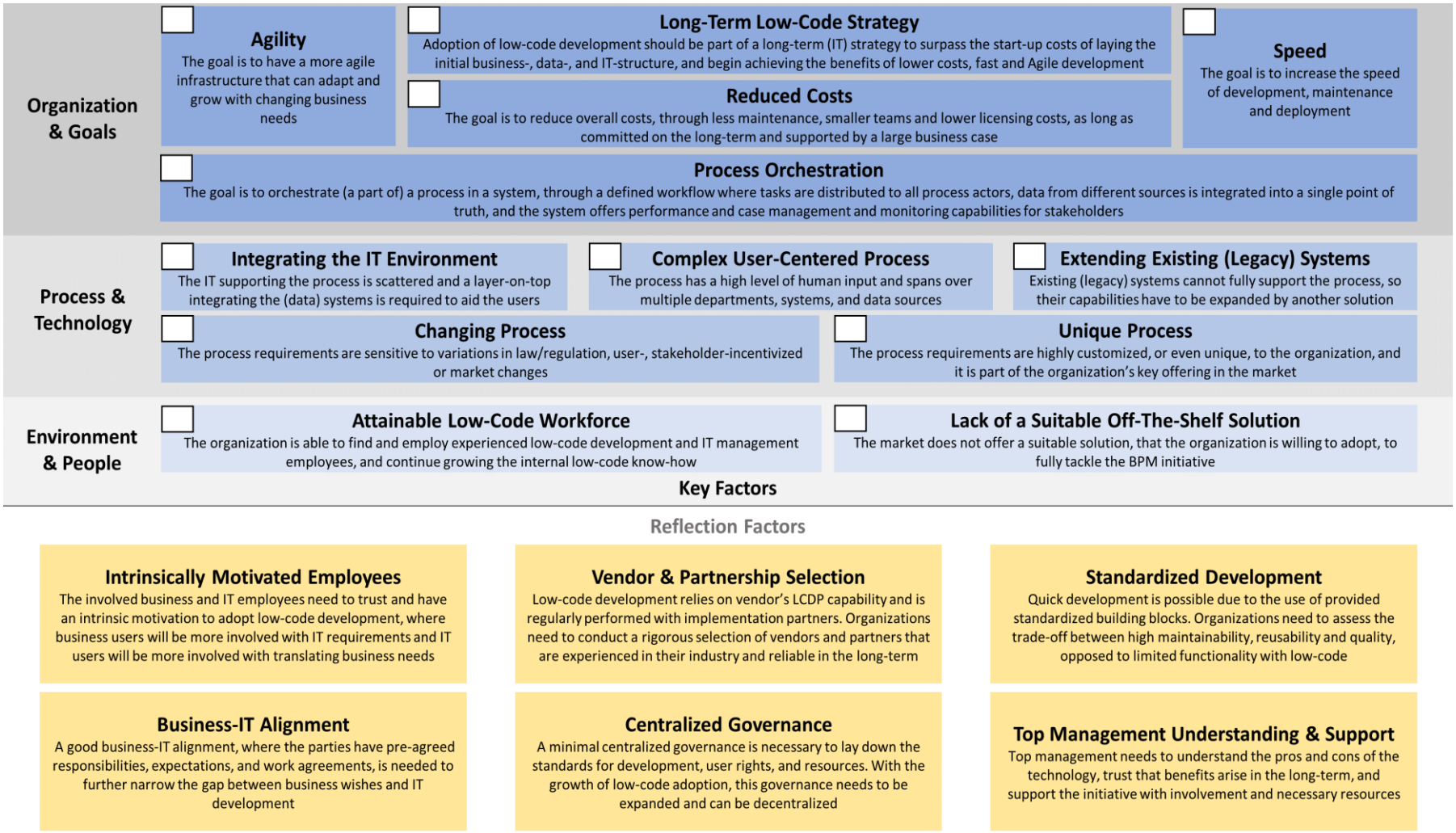

Figure 2. Low-code BPM assessment framework.

Agility (Key) – Besides speed, a key reason for low-code adoption is the increase in organizational responsiveness to changes (Alsaadi et al., 2021; Frank et al., 2021; Cai et al., 2022). Organizations have employed low-code BPM "*where process changes are expected quickly, for example, legislative changes. Or where the consumer demands something different all the time*" (I4), but low-code also allows to "*build a system that evolves with you, grows with changing business*" (I2). Business users can be involved in application management which allows organizations to adjust to changing market conditions more easily (Olariu et al., 2016). Therefore, low-code development is suitable for organizations that require high levels of agility in their IT landscape. An important reflection, concerning speed and agility, is the effects of standardized development discussed further.

Reduced Costs (Key) – Cost-effectiveness is also possible through the employment of low-code in the long-term (Luo et al., 2021; Cai et al., 2022). Through the reuse of proven components, each subsequent project has the potential to reduce overall costs (Cicman et al., 2021). I6 suggested: "*If you have a lot of component reuse, you can develop your new application faster and faster, that is cheaper than coding an application each time […] Because you need fewer people, you have lower licensing costs, less maintenance, et cetera*". To become cost-effective, low-code development should be part of a long-term IT strategy where the organization continues actively developing using low-code, experts state. I8 explained: "*As you start scaling up and getting a few apps, you start seeing economies of scale. Essentially, it is like I am using one platform to do multiple things, so I'm not paying for a new license each time*". Organizations seeking lower IT development costs can consider low-code as a solution.

4.2 Organizational dimension factors

This dimension includes organization-wide characteristics that are essential with low-code BPM.

Long-Term Low-Code Strategy (Key) – MDE adoption literature shows that organizations working on a project-by-project basis unjustly disregard the technology while organizations with a progressive and iterative project approach succeed (Hutchinson et al., 2014; Whittle et al., 2017). Experts, such as I6, also emphasize this: "*Building one application with low-code is really a total waste of money, you really need to have a strategy […] Only when you get to 5/6 applications, then you suddenly see the added value of low-code*". Not only the benefits depend on long-term adoption, as I3 illustrated: "*If you train people in certain tooling, it only makes sense if that is on the long-term, if there's a long-term vision there. Or you say, we'll bring in a partner with whom we'll do a partnership to make it a success. But still, in a partnership, you also only do that for the long-term*". Therefore, it is key that organizations incorporate low-code in a long-term strategy used for, possibly, multiple solutions in the future.

Business-IT alignment (Reflection) – Lacking business-IT alignment is seen as a deep-rooted obstacle for any new technology implementation in an organization (Luftman & Brier, 2019). Even further, good Business-IT alignment has been essential for MDE adoption as it is often a business decision, where the business- and IT goals are to be aligned (Whittle et al., 2017). Multiple experts agreed to the importance of business-IT alignment, especially in the starting phase as I6 illustrated: "*It is a cultural aspect, you are suddenly going to put two cultures together and say, well, let's work together. You have to guide that well*". Apart from its presence prior to adoption, low-code development in an organization gives the "*ability to involve the rest of the organization in the development process*" (I1), raising the business-IT alignment. This was a reason for adopting low-code BPM in I1's organization. Organizations ought to assess themselves on whether their business-IT alignment is suitable for low-code BPM adoption.

Centralized Governance (Reflection) – IT governance entails the guidance and structures in place to sustain the IT infrastructure in enabling an organization's strategy and objectives (de Haes & van Grembergen, 2009). Proper governance is deemed important in BPM initiatives (vom Brocke & Schmiedel, 2014) as with MDE adoption in organizations (Aranda et al., 2012) to result in long-term benefits for an organization. Most experts agreed that a certain level of governance is needed, as I3 explained: "*If we start with a team or within a department, it is a bit of an overkill to set up a dedicated device like that […] but if you let everyone onto the platform, you get uncontrolled growth, so it is crucial to set-up certain [development] standards*". Hoogsteen and Borgman (2022) argued that a centralized governance structure is the most suitable form for low-code development. Experts expanded

on this notion, as I4 stated: "*In the starting phase: centralized, standards, laying the foundation. When building onwards, then decentralized is really perfect*". Therefore, organizations should be aware that implementing low-code BPM entails setting up a centralized governance structure in the organization.

Top Management Understanding & Support (Reflection) – Especially in the case of technology adoption, top management support is vital (Hsu et al., 2019). Active top management support includes, for example, communication of opportunities or proactive guidance through technology training as opposed to pure decision-making with passive support (Hoogsteen & Borgman, 2022). However, various experts explained that organizations struggle to fully grasp the technology and how efforts can be translated into results, as I7 explained: "*In the beginning, you will not notice anything. Because what are you doing? You are putting a layer on top of what you already had. It does not give any new functionality*". Therefore, I6 argued that understanding is key: "*The board has to be along too, they have to understand why it is important, and that it takes time, and it will not be so quick in the beginning*". Therefore, this factor accentuates the understanding and support that the top management should have on how the cited benefits of low-code come about.

4.3 Process dimension factors

All process-related characteristics suitable for low-code BPM support are discussed in this dimension.

Complex User-Centered Process (Key) – BPMSs support modeling processes exactly as they are performed (Ravasan et al., 2014). Process selection is, therefore, key as organizations can endlessly map business processes. Dumas et al. (2018) proposed various criteria including the process size where at least three different actors have to be involved. A certain level of process complexity is mentioned by FG3 as key for low-code BPM: "*Longer chain processes spanning multiple departments are often a good indicator for high [low-code BPM] potential. Because those kinds of processes are tough to manage*". Experts agreed that the full potential of low-code BPM is unlocked with such complex processes. Moreover, I7 emphasized the user, and how low-code BPM imposes a standardized, clear way of sharing data, in such complex processes: "*If the complexity is that you have many different parties or components that do not communicate well with each other, that use Excel exports […] low-code could indeed possibly be the solution for you*". Therefore, organizations orchestrating complex, user-centered processes are well suited to low-code BPM.

Unique Process (Key) – Make-or-buy literature states that core processes are better supported by acquired packaged software while strategic systems should be developed in-house (McManus, 2003). In line herewith, low-code's customizability is especially useful and becomes cheaper than adapting packaged solutions (Bratincevic, 2020), for unique processes. Experts concurred with the above, as FG1 explained: "*we see that [low-code is used when] the market cannot fulfill all the [process] requirements […] you buy standard applications for your HR or CRM processes, which is much cheaper*". Uniqueness, however, is often hard to measure objectively. I2, therefore, further emphasized uniqueness: "*The system […] has to be an integral part of that business, that the system should help that customer differentiate itself from the rest of the market*". Low-code BPM is, therefore, ideal for highly customized business processes, part of the organization's key offering.

Changing Process (Key) – Low-code BPM provides a flexible modeling approach that, together with development speed, allows constant change in the process (Koplowitz, 2017). Therefore, processes that have a tendency to vary become compelling for low-code BPM. FG1 saw in his own experience: "*If a lot of flexibility is needed in the process or there can be a lot of changes in process requirements, then you can swiftly use low-code*". I6 drew a comparison to non-changing processes: "*Is it a solution that is in your core process and does not change much? Then I would take a SaaS solution or an ERP solution, or at least a standard solution*". All in all, apart from complex and unique processes, low-code BPM suits organizational processes that are sensitive to changes.

4.4 Technical dimension factors

This dimension discussed technical considerations and characteristics specific for low-code BPM.

Integrating the IT Environment (Key) – BPMS compatibility with the surrounding IT infrastructure is essential for successful implementation (Bosilj Vukšić et al., 2018). Low-code BPM, as the 'process orchestration' factor showed, allows to flexibly integrate (data)systems underlying the process (Sahay et al., 2020; Frank et al., 2021) to create an orchestration layer over the IT landscape. I7 explained how this helps organizations to manage their process: "*Low-code BPM is a layer over your systems […] where you can pull information from different systems, forming it into a unified process in that one layer which means you have a lot more control over how that process goes. […] If something changes in the process, or it does not work, or you want to improve something, you do that in the above layer*". Therefore, I8 sees low-code being applied in: "*An organization where they're a very decentralized, they've got systems all over the place, they have a very scattered landscape […] Having one platform in which they can connect everything, to some degree, is very powerful*". Hence, low-code BPM is a solution for organizations with a scattered IT landscape, supporting the process which needs integration.

Extending Existing (Legacy) Systems (Key) – In a report by OutSystems (2019), a common use of low-code was found to be extending existing systems' functionality. As low-code BPM serves as a layer-on-top of your IT infrastructure, functionality can be added in that layer as well. FG2 recounted a client case: "*You can use it as a custom layer on top of large packaged solutions, in SAP where you do not want to customize due to complexity and costs. […] You can add some extra process digitalization if it is not possible in a packaged solution*". Moreover, Sahay et al. (2020) argued low-code being used for the integration and extension of legacy systems as well. This allows to communicate and use the functionality of legacy systems without making any adjustments in the settings. I4 explained: "*With large companies, you have a choice: Am I going to remove the legacy systems? Then you are going to have to convert which are expensive projects. But you can also offer a solution through low-code where you make Appian communicate with your legacy. That is easy to do and you can let your legacy be as it is*". Therefore, if organizations have systems that they do not want to adjust for the sake of process orchestration, low-code provides an unintrusive way of communicating with (legacy) systems.

Standardized Development (Reflection) – I5 summarized low-code development as: "*The disadvantage is also very much an advantage. […] So you use certain standardized blocks and you link them together. Because of the blocks, you are limited in what you can do, which also means you have a fewer bugs. So you have a fewer problems to solve but it does limit what you can and cannot do*". On one side, concerns about the ability of LCDPs have previously deterred adoption (Sanchis et al., 2020; Alsaadi et al., 2021; Luo et al., 2021). On the other side, experts argued that maintainability, reusability, and quality are increased due to the use of standardized blocks. I4 summarized these benefits through the following analogy: "*You can imagine it is much easier to build a castle with Lego blocks than with sand. Because those low-code blocks have great quality, you can hardly make any mistakes*". Organizations should, thus, assess if their software development requirements are met by the LCDP.

4.5 Social dimension factors

These factors concern employees and middle-managers responsible for implementing low-code BPM.

Attainable Low-Code Workforce (Key) – Although low-code vendors promote that any business user can develop on their platform, low-code development still requires a technical background (Frank et al., 2021; Hoogsteen & Borgman, 2022) and has a steep learning curve (Luo et al, 2021). In the meantime, experts argued that experienced know-how is essential when implementing low-code. I6 illustrated the current problem with this: "*There are not that many people who have enough experience. […] Those that have are very expensive and rare. […] The whole market is still catching up in that respect*" (I6). All experts that directly led low-code implementations (FG2, I1, I2, I3, I4, I6) emphasized the difficulties in finding experienced employees. For example, I4 mentioned: "*Suppose you start using Appian in the Netherlands, there are no good people left to get because they have now been bought up by [company X], [company Y], and others*". Moreover, it does not only concern developers, but also solution architects and business analysts. Being able to attain a low-code workforce is a key decision-making point when considering low-code BPM.

Intrinsically Motivated Employees (Reflection) – MDE studies have shown that developers mistrusting the technological capabilities employed work-arounds, hereby defeating the purpose of the technology (Hutchinson et al., 2014), or reverted to traditional development (Aranda et al., 2012). Mistaken beliefs in BPMS capabilities previously caused fear of job loss, dissatisfaction, and conflicts among employees (Bach et al., 2017). The attitude of both business users and developers is also important for low-code BPM adoption. Regarding developers, I1 explained: "*you cannot dump a completely new tooling on your Java-group, that just does not work*", while regarding business users, I6 emphasized: "*Business is going to have to put a lot more time in specifying and prioritizing user stories […] if they do not want to put the time in that, than it still is not going to work*". It is important organizations assess for themselves how their employees foresee the implementation of low-code BPM.

4.6 Environmental dimension factors

This dimension relates to topics, relevant to low-code BPM, that are outside of the boundaries of the organization.

Lack of a Suitable Off-The-Shelf Solution (Key) – McManus (2003) stated: "*Executives agree that certain unique business applications will necessitate creating software in-house, regardless of time or cost considerations*". Low-code's customizability offers to cater to such unique business applications (Alsaadi et al., 2021). In our scope, a unique business application can be needed to suit a 'unique process', as seen with an earlier factor. However, albeit a solution for the process exists, I4 explained: "*Or you just do not want to accept a solution. Even if the solution fits for 80/90%, then I would still not consider it suitable*". Although the percentages are subjective, a lack of a suitable off-the-shelf solution can be a key indicator for low-code BPM adoption. I1 explained how this played a role in an organization choosing low-code: "*The organization spent a very long time looking for a package for settling claims and to get a handle on the backend of that process. They did not find anything for that*". Therefore, a lack of a suitable off-the-shelf solution is a key indicator that low-code BPM is a beneficial solution.

Vendor & Partnership Selection (Reflection) – BPMS literature highlights the vendor's reputation, knowledge, and experience in serving clients as an important contributing factor (Bosilj Vukšić et al., 2018). As organizations are dependent on the vendor's platform, and fear vendor lock-in (Alsaadi et al., 2021), a careful vendor selection should be conducted (Cai et al., 2022). Moreover, an implementation partner is often proposed for their knowledge, as I5 explained: "*you do need someone with knowledge looking over your shoulder […] if you have to do it all yourself, if you have to reinvent the wheel, it will take a long time*" (I5). Moreover, FG2 emphasized this importance in the scope of low-code BPM: "*We now really focus on low-code BPM tooling or platforms, which is, in my experience, actually always implemented through external partners*" (FG2). Organizations ought to realize the importance behind rigorous selection of vendor and implementation partnerships when considering low-code BPM.

4.7 Visual design of the framework

The framework went through three design iterations to come to the final version. In the process, various design decisions have been taken, further explained in this section.

The focus group showed how different factor types used interchangeably caused confusion. By segregating the factors based on factor type, we integrated structure into the framework. Users can now interpret each 'part' of the framework in their own way. Moreover, multiple experts proposed using the framework as a 'tick-off' list. I2 proposed: "*Yes, you can check off on that. I think it would be nice for a lot of my clients to have*" and I7 envisioned: "*Look, if I made it simple, this would be a very nice checklist […] for when we go to clients. So do you meet these points, is this indeed approximately what you want with key [factors]? We have reflection, well keep in mind that within the organization so you do have to start arranging these kinds of things*". Therefore, white boxes are added to the key factors as 'tick-off' boxes. Lastly, we have decided not to explicitly include the dimensions in the framework. Initially, two variations of the final framework were made with and without color-coded dimensions. The researchers and various colleagues agreed that the simplified framework was easier to perceive and

the dimensions did not add much to the framework. Moreover, the global 'themes' on the right still provide some context on the factor's origin.

5 Framework evaluation

We evaluated the framework on the criteria posed in Section 3.1 with five fictitious use case simulations. In each simulation, a participant evaluated a fictitious use case's suitability for low-code BPM.

Historical, real-life use cases from a consultancy firm were used for these simulations. Prior, the manager of each use case explained the business problem that the organization had faced and a use case description was formulated together with the researchers. As the participants were consultants in the same firm, each description was anonymized, so it could not be traced back to the actual projects. During each simulation, we introduced the low-code BPM assessment framework and the use case description to the participant. Moreover, the participant could interview the manager for more in-depth insights on the use case. The participants assessed each fictitious use case's suitability for low-code BPM using the framework. After the simulation, each participant gave their assessment, its rationale, experience while using the framework, and the participants filled a survey measuring the evaluation criteria on a 1 to 5 scale. This section presents how the framework performed during the fictitious use case evaluations.

Completeness was rated highly by the five participants with an average score of 4.4/5. The participants agree that the framework's dimensions cover the general topics that they would consider when assessing a use case. Moreover, one participant explicitly emphasized the broad spectrum of questions that can be asked using the framework and: "*On the basis of this, I think you can give an excellent opinion whether or not it fits the organization at first glance*". Another participant proposed to utilize this model as a theoretical model, whereafter specific factors are further explored in an organization's case. In general, participants agreed that it is a complete framework for an initial use case assessment.

Internal consistency scored lower with a 3.7/5. The factor definitions, and how the factors should be interpreted through key- and reflection factor types, were understood and accepted by the participants. However, participants struggled with some factors overlapping in definition, such as 'Agility' and 'Changing Process'. Whereas the first focuses on an organizational goal and the latter on a process characteristic, we agree that these could be confusing. Lastly, some participants questioned how specific certain factors were to low-code BPM. Although it is true that some factors, such as 'Business-IT alignment', are more generally applicable to any adoption of technology, we argue that the substantiation of each factor explains why, and how, the factor is relevant in the context of low-code BPM.

Operationality received an average of 4.2/5 from the participants. Participants see the framework as a useful fundamental theory during, or when preparing for, an interview. Some used the tick-off boxes for this purpose, while others added plus- and minus signs. One participant noted the following regarding his approach: "*I briefly grabbed that framework during the interview to see like, okay, do I have all the points? Did I walk through it all? Then I know for sure that I can give good advice instead of doing it on gut feeling*". Moreover, participants noted that the difference in factor types allows them to prioritize the assessment and, possibly, initially omit reflection factors. Nevertheless, some participants argued that the framework leaves much to interpretation which makes it too subjective to do a thorough assessment. An often-recurring theme was the quantification of factors, explained in the next paragraph.

Ease of use scored a bit lower than operationality, with a 3.9/5. Predominantly, the participants struggled to fully grasp the framework and perform an assessment in the 30-minute session. Consequently, some factors or other aspects were misinterpreted. One participant shared: "*Having done it once already, I can also much more easily think of sub-questions. Now, sometimes I got stuck*". Therefore, instead of an ad-hoc framework that can be used instantly, this framework is more suitable for careful studying of a use case. More objective quantification of factors could further elevate this framework. Multiple participants asked how a missing factor or how the factor weight should be interpreted. Future research could find specific indicators of how an organization is performing at a certain factor. This would help in interpreting the framework but also in making it more operational to do an actual assessment.

Concluding from the evaluation, we argue that our framework is useful for an initial judgement of a use case's suitability for low-code BPM. However, it would help users to make the factors less interpretable and specify how factors are related to each other to allow a more rigorous and objective assessment.

6 Discussion and conclusion

In times when organizations are urgently looking to digitize, optimize, and automate their business processes, low-code BPM emerges as an effective solution. Low-code vendors advertise the easiness of their product and potential benefits while scientific literature nuances these benefits with concerns on customization, vendor lock-in, and complexity. Meanwhile, organizations struggle to identify the right solutions to advance their business processes, but existing literature does not provide guidance on assessing whether low-code BPM can be that solution. To solve this gap, we proposed the low-code BPM assessment framework. It consists of 18 factors to be assessed by organizations regarding their BPM initiative and illustrates important considerations when deciding on low-code BPM adoption.

The scientific contribution of our study is twofold. Firstly, our study is one of the first to research low-code in the context of BPM. Cai et al. (2022) proposed a method for process automation through use of an LCDP. However, their method assumes the low-code decision to have been made and focuses on a small-scale automation. Still, Luo et al. (2020) found the process automation and BPM domains among the most-used in practice, making this a notably unresearched area. Therefore, our study contributes by forming a scientific understanding of this application and we encourage researchers to use and extend this scientific base when studying low-code BPM. Secondly, our study extends current knowledge on the organizational use of low-code. Various studies have been performed on low-code's features (Sahay et al., 2020; Frank et al., 2021; Luo et al., 2021) and to elicit the low-code adoption reasons (Sanchis et al., 2020; Alsaadi et al., 2021). But none describe the criteria for effective organizational use of low-code (Frank et al., 2021). Our framework presents these criteria through 18 factors in the BPM context.

Our study also has practical contributions. Firstly, the framework can be utilized by organizations to assess the applicability of low-code BPM for their processes. The evaluation showed that the framework serves as a theoretical map to understand which elements are key and which need to be reflected upon. Secondly, implementation specialists and consultants can use the framework to prepare and validate a more thorough assessment of an organization's BPM case leading to an IT adoption decision.

This study has several limitations. First, the reliability of qualitative research in evaluating information systems decreases due to the subjectivity of researchers' interpretations (Kaplan & Maxwell, 2005). To combat this threat to reliability, we have used both scientific and grey literature from various domains, and employed various empirical analyses to triangulate the data gathered throughout this study. We argue that this represents a credible picture of significant low-code BPM factors. Secondly, the concept low-code has never clearly been defined, making the study of a specific type of LCDP potentially inconclusive. Interviewees could misinterpret the object of study, threatening construct validity. Therefore, we have conceptualized our definition of low-code BPM based on existing literature and communicated it to all participants in this study. Lastly, the framework's evaluation consisted of a rather small sample of five participants. Although a larger evaluation sample size could have been more fruitful, the comments made by the participants became increasingly identical, indicating theoretical saturation. Moreover, earlier comments by experts coincided with the evaluation results.

We see several directions for further research. First, the reliability of the framework could be increased by carrying out evaluations at different types of organizations. It would be especially interesting to target young start-ups, facing an IT adoption decision, since organizational size has been cited to affect the LCDP adoption decision (Sahay et al., 2020). Third, to improve the operational potential of this framework, future studies should focus on providing each factor with measurements. Organizations would, then, be able to rely on object factor measurements, instead of subjective analyses of their BPM initiative. Finally, future research may focus on the adoption criteria of other LCDPs outside of the BPM scope. Our study is a first step to help understand and recognize when low-code can advance organizations further. We encourage to use and extend this knowledge in further developing low-code understanding.

References

Alsaadi, H. A., Radain, D. T., Alzahrani, M. M., Alshammari, W. F., Alahmadi, D., and Fakieh, B. (2021) 'Factors that affect the utilization of low-code development platforms: survey study', *Revista Română de Informatică Și Automatică*, 31(3), 123–140.

Aranda, J., Damian, D., and Borici, A. (2012) 'Transition to model-driven engineering', *In: International Conference on Model Driven Engineering Languages and Systems*, pp. 692-708.

Bach, M. P., Vukšić, V. B., and Vugec, D. S. (2017) 'Individual's resistance regarding BPM initiative: case study of the insurance company', *Naše gospodarstvo/Our economy*, 63(4), 29-39.

Bock, A. C. and Frank, U. (2021a) 'Low-code platform', *Business & Information Systems Engineering*, 63(6), 733-740.

Bock, A. C. and Frank, U. (2021b) 'In search of the essence of low-code: an exploratory study of seven development platforms', *In: 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 57-66.

Bosilj Vukšić, V., Brkić, L., and Tomičić-Pupek, K. (2018) 'Understanding the success factors in adopting business process management software: Case studies', *Interdisciplinary Description of Complex Systems: INDECS*, 16(2), 194-215.

Bratincevic, J. (2020). *When and How To Modernize Core Applications Using Low-Code Platforms*. Available at: https://www.forrester.com/report/When-And-How-To-Modernize-Core-Applications-Using-LowCode-Platforms/RES155943.

Braun, V., and Clarke, V. (2006) 'Using thematic analysis in psychology', *Qualitative research in psychology*, 3(2), 77-101.

Brkić, L., Tomičić Pupek, K., & Bosilj Vukšić, V. (2020). A Framework for BPM Software Selection in Relation to Digital Transformation Drivers. *Tehnički vjesnik*, 27(4), 1108-1114.

Cabot, J. (2020) 'Positioning of the low-code movement within the field of model-driven engineering', *In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 1-3.

Cai, F. Z., Huang, S. Y., Kessler, T. S., and Fottner, F. J. (2022) 'A Case Study: Digitalization of Business Processes of SMEs with Low-Code Method', *IFAC-PapersOnLine*, 55(10), 1840-1845.

Cicman, J. and Bratincevic, J. (2021) *Forget About Build Versus Buy; Your Choice Is Customize Or Compose*. Availabe at: https://www.forrester.com/report/forget-about-build-versus-buy-your-choice-is-customize-or-compose/RES162576.

Crabtree, B. and Miller, W. (1999) 'A template approach to text analysis: Developing and using codebooks'. *In:* B. Crabtree & W. Miller, eds., *Doing qualitative research*, pp. 163–177.

De Haes, S., and Van Grembergen, W. (2009) 'An exploratory study into IT governance implementations and its impact on business/IT alignment', *Information Systems Management*, 26(2), 123-137.

Denner, M. S., Püschel, L. C., and Röglinger, M. (2018) 'How to exploit the digitalization potential of business processes', *Business & Information Systems Engineering*, 60(4), 331-349.

Dumas, M., Rosa, M. L., Mendling, J., and Reijers, H. A. (2018). *Fundamentals of Business Process Management*, 2nd ed., Berlin, Germany: Springer.

Frank, U., Maier, P., and Bock, A. (2021) 'Low code platforms: Promises, concepts and prospects. A comparative study of ten systems', *ICB-Research Report*, No. 70.

Hevner, A. R., March, S. T., and Park, J. (2004). 'Design science in information systems research', *MIS Quarterly*, 28(1), 75–105.

Hoogsteen, D., and Borgman, H. P. (2022) 'Empower the Workforce, Empower the Company? Citizen Development Adoption', *In: Proceedings of the 55th Hawaii International Conference on System Sciences*, pp. 1-10.

Hsu, H. Y., Liu, F. H., Tsou, H. T., and Chen, L. J. (2019) 'Openness of technology adoption, top management support and service innovation: a social innovation perspective', *Journal of Business & Industrial Marketing*, 34(3), 575-590.

Hung, P. and Low, G. C. (2008) 'Factors Affecting the Buy vs Build Decision in Large Australian Organisations', *Journal of Information Technology*, 23(2), 118–131.

Hutchinson, J., Whittle, J., and Rouncefield, M. (2014) 'Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure', *Science of Computer Programming*, 89, 144–161.

Iho, S., Krejci, D., and Missonier, S. (2021), 'Supporting Knowledge Integration with Low-Code Development Platforms', *In: ECIS 2021 Research Papers*, 38.

Konopik, J., Jahn, C., Schuster, T., Hoßbach, N., and Pflaum, A. (2022) 'Mastering the digital transformation through organizational capabilities: A conceptual framework', *Digital Business*, 2(2), 100019.

Koplowitz, J. R. (2017) *Vendor Landscape: BPM Platforms For Digital Automation*. Available at: https://www.forrester.com/report/vendor-landscape-bpm-platforms-for-digital-automation/RES128401.

Krueger, R. A. and Casey, M. A. (2014) *Focus groups: a practical guide for applied research*, 4th ed., Thoasand Oaks, California: Sage.

Luftman, J. and Brier, T. (1999) 'Achieving and sustaining business-IT alignment', *California management review*, 42(1), 109-122.

Luo, Y., Liang, P., Wang, C., Shahin, M., and Zhan, J. (2021) 'Characteristics and Challenges of Low-Code Development', *In: Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1-11.

Mamudu, A., Bandara, W., Wynn, M.T. and Leemans, S.J. (2022) 'A Process Mining Success Factors Model', *In: International Conference on Business Process Management*, pp. 143-160.

March, S. T. and Smith, G. F. (1995) 'Design and natural science research on information technology', *Decision support systems*, 15(4), 251-266.

McManus, D. J. (2003) 'A Model of Organizational Innovation: Build versus buy in the decision stage', *The International Journal of Applied Management and Technology*, 1(1), 29-44.

Metrôlho, J. C. M. M., Araújo, R., Ribeiro, F., and Castela, N. (2019) 'An approach using a low-code platform for retraining professionals to ICT', *In: Proceedings of EDULEARN*, 7200-7207.

Meuser, M., and Nagel, U. (2009) 'The expert interview and changes in knowledge production', *In: Interviewing experts*, pp. 17-42.

Moran-Ellis, J. et al. (2006) 'Triangulation and integration: processes, claims and implications', *Qualitative research*, 6(1), 45-59.

Morgan, D. (1995) `Why things (sometimes) go wrong in focus groups', *Qualitative Health Research*, 5 (4): 516-523.

Olariu, C., Gogan, M. and Rennung, F. (2016) 'Switching the Center of Software Development from IT to Business Experts Using Intelligent Business Process Management Suites' *In:* Balas, V., Jain, L.,

Kovačević, B. (eds) *Soft Computing Applications. Advances in Intelligent Systems and Computing*, 357. Springer, Cham.

OutSystems (2019) *The State of Application Development - Is IT Ready for Disruption?*, Outsystems, Available at: https://www.outsystems.com/local-gov/-/media/053D5BCC32364C2993C8D0BAFA880DB1.ashx.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007) 'A design science research methodology for information systems research', *Journal of management information systems*, 24(3), 45-77.

Plattfaut, R., Borghoff, V., Godefroid, M., Koch, J., Trampler, M., and Coners, A. (2022) 'The Critical Success Factors for Robotic Process Automation', *Computers in Industry*, 138, 103646.

Prat, N., Comyn-Wattiau, I., and Akoka, J. (2015) 'A taxonomy of evaluation methods for information systems artifacts', *Journal of Management Information Systems*, 32(3), 229-267.

Rands, T. (1993) 'A framework for managing software make or buy', *European Journal of Information Systems*, 2(4), 273-282.

Ravasan, A. Z., Rouhani, S., and Hamidi, H. (2014) 'A Practical Framework for Business Process Management Suites Selection Using Fuzzy TOPSIS Approach', *In: Proceedings of the 16th International Conference on Enterprise Information Systems, pp.* 295-302.

Ravesteyn, P. and Batenburg, R. (2010) 'Surveying the critical success factors of BPM-systems implementation', *Business Process Management Journal*, 16(3), 492–507.

Richardson, C. and Rymer, J. R. (2014) *New development platforms emerge for customer-facing applications*. Available at: https://www.forrester.com/report/New-Development-Platforms-Emerge-For-CustomerFacing-Applications/RES113411.

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

Sadovnikov, K. (2022). Conceptual Framework Assessing a BPM Initiative for Low-Code Development Platform Suitability [Master's thesis, Utrecht University]. Available at: https://studenttheses.uu.nl/handle/20.500.12932/43359

Sanchis, R., García-Perales, S., Fraile, F., and Poler, R. (2020) 'Low-Code as Enabler of Digital Transformation in Manufacturing Industry', *Applied Sciences*, 10(1), 12.

Van Veldhoven, Z. and Vanthienen, J. (2022) 'Digital transformation as an interaction-driven perspective between business, society, and technology', *Electronic Markets*, 32, 629-644.

Vincent, P. *et al.* (2020) *Magic quadrant for enterprise low-code application platforms*. Available at: https://www.gartner.com/en/documents/3991199.

Vom Brocke, J., Zelt, S., and Schmiedel, T. (2016) 'On the role of context in business process management', *International Journal of Information Management*, 36(3), 486-495.

Vom Brocke, J., Zelt, S., and Schmiedel, T. (2016) 'On the role of context in business process management', *International Journal of Information Management*, 36(3), 486-495.

Waszkowski, R. (2019) 'Low-code platform for automating business processes in manufacturing', *IFAC-PapersOnLine*, 52(10), 376–381.

Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., and Heldal, R. (2017) 'A taxonomy of tool-related issues affecting the adoption of model-driven engineering', *Software & Systems Modeling*, 16(2), 313-331.

Xu, L. D., Xu, E. L., and Li, L. (2018) 'Industry 4.0: state of the art and future trends', *International journal of production research*, 56(8), 2941-2962.