MIS Quarterly Executive

Volume 23 | Issue 3 Article 7

September 2024

How GuideCom Used the Cognigy.AI Low-Code Platform to Develop an AI-Based Smart Assistant

Imke Grashoff

Jan Recker

Recommended Citation

Grashoff, Imke and Recker, Jan (2024) "How GuideCom Used the Cognigy.AI Low-Code Platform to Develop an AI-Based Smart Assistant," *MIS Quarterly Executive*: Vol. 23: Iss. 3, Article 7.
Available at: https://aisel.aisnet.org/misqe/vol23/iss3/7

MIS QUARTERLY EXECUTIVE

How GuideCom Used the Cognigy.AI Low-Code Platform to Develop an AI-Based Smart Assistant

GuideCom, a medium-sized German software provider, adopted a low-code approach to develop its first AI product. GuideCom made the strategic choice to use Cognigy.AI, a low-code development platform, which both enabled and constrained the development of an AI-based chatbot. This case study shows that low-code AI development platforms make it easier to integrate different expertise required for AI development. However, firms need to consider constraints related to expandability, security and autonomy issues when deciding to adopt a low-code approach for AI development.[1,2]

Imke Grashoff
University of Hamburg (Germany)

Jan Recker
University of Hamburg (Germany)

The Role of Low-Code Development Platforms in Building AI-Based Business Solutions

The increasing use of artificial intelligence (AI) in business and throughout society is prompting organizations of all kinds and sizes to assess how they can use AI to stay competitive. Their decision-making must be both strategic and fast. In fact, a tipping point has been reached: Firms no longer need to decide *whether* to adopt AI; instead, they need to know how to adopt AI. Small and medium-sized firms, in particular, need to evaluate carefully which strategy for AI adoption to choose, because they often have limited resources and expertise.

Depending on their goals and available resources, firms have three choices when implementing AI solutions. First, they can buy and customize an off-the-shelf AI solution. This approach does not require advanced experience with AI or substantial resources to develop AI for a specific task, but it does not enable individualized solutions to be created. Second, they can develop a proprietary AI solution independently in-house, provided they have the required resources (such as AI engineers and data scientists). This approach enables companies to tailor an AI solution specifically to their purposes, but such an endeavor is complex, time-consuming and depends on expert knowledge.[3] Third, companies can choose a middle path between customization and in-house development: They can follow a low-code approach to the development of AI-based business solutions.

[1] Noel Carroll is the senior accepting editor for this article.

[2] The authors thank Noel Carroll and the members of the review team for their thoughtful feedback and guidance throughout the review process. We also gratefully acknowledge the cooperation of GuideCom and Cognigy.AI, both of which allowed us to partner in their project.

[3] van den Broek, E., Sergeeva, A. and Huysmann, M. "When the Machine Meets the Expert: An Ethnography of Developing AI for Hiring," *MIS Quarterly* (45:3), September 2022, pp. 1557-1580.

DOI: 10.17705/2msqe.00101

Low-code development platforms use declarative, high-level programming abstraction and model-driven development principles to hide code complexity behind intuitive graphical user interfaces, which are used for workflow development, business logic specification and data engineering. They provide a range of instantly available software components, such as web services, interface widgets and connectors, that enable faster program composition. They promise to lower the skill barrier to software development in firms by allowing tech-savvy but coding-inexperienced users to program their own software applications. Such platforms can thus provide a remedy for the common lack of IT specialists because they speed up a firm's ability to develop or adapt software and free up IT resources for more complex tasks.[4]

Low-code development platforms originated in the context of conventional software development but also promise to make AI development simpler, cheaper and more accessible. Some researchers claim that low-code development platforms for AI development hold great potential for "the democratization of AI."[5] They report that low-code AI development platforms can even be used by small- and medium-sized firms that usually do not have the financial, human or IT resources required to independently develop their own AI solutions.

Nevertheless, low-code AI development platforms are not a strategic panacea. Using them poses several challenges, such as the need for a robust implementation strategy, limited architectural flexibility and preparatory data work[6] (i.e., collecting and preparing the data needed to train the AI model). Thus, potential adopters of low-code AI development platforms need to consider whether such platforms will actually *enable* AI development by providing easier and faster access to interface, data and workflow development, or *constrain* AI development by hiding the required complexity in architecture or by overlooking relevant data work.

In this article, we explore the role of low-code AI development platforms to better understand their enabling and constraining effects. Knowledge about the impacts of the enablers and constraints will be important to CIOs and other technology executives as they formulate an AI implementation strategy and evaluate the potential, risks and prerequisites of a low-code approach to AI development. This knowledge will also help those involved in the actual implementation projects, who must find effective ways to use low-code development tools when implementing AI with limited resources.

Our findings in this article are distilled from a case study of GuideCom, a medium-sized German software development company that over 16 months in 2022 and 2023, developed a new AI-based product offering based on the Cognigy.AI low-code development platform for enterprise conversational AI automation. The product is a conversational chatbot system for the automation of HR service processes and was developed in conjunction with one of GuideCom's clients, a major German bank. We followed the entire development project from conception to implementation and collected data through observations, formal interviews, informal conversations and document analysis. Our research process is described in the Appendix.

Before describing the GuideCom case, we provide a brief tutorial on low-code software development and AI development to understand the intersection of both approaches. We then examine GuideCom's conversational chatbot system development process, which comprised the following stages: perceiving AI as a new technology; evaluating available AI market offerings and providers; identifying a suitable low-code platform as the technological foundation; defining requirements and use cases; collecting training data; prototyping, building, training and testing the model; and implementing the system with the client. From the lessons learned at GuideCom, we present a conceptual framework that visualizes low-code AI development as a process in which the conflicts between the interdependent enablers and constraints inherent in a low-code AI

[4] Matook, S., Wang, Y. M, Koeppel, N. and Guerin, S. "Metacognitive Skills in Low-Code App Development: Work-Integrated Learning in Information Systems Development," *Journal of Information Technology* (39:1), March 2024, pp. 41-70.

[5] Sundberg L. and Holmström, J. "Democratizing Artificial Intelligence: How No-Code AI Can Leverage Machine Learning Operations," *Business Horizons* (66:6), 2023, pp. 777-788.

[6] Alamin, M. A. A., Uddin, G, Malakar, S. and Afroz, S. "Developer Discussion Topics on the Adoption and Barriers of Low Code Software Development Platforms," *Empirical Software Engineering* (28:1), January 2023.

development platform approach can be resolved. Finally, we provide practical recommendations for using low-code AI development platforms to build AI-based business solutions.

Understanding the Complementary Characteristics of Low-Code Development Platforms and AI Development

Below, we provide a brief tutorial on low-code software and AI development and show how the characteristics of each can be complementary.

Low-Code Software Development

Driven by the pressure of continuous digitalization, firms are experiencing an increasing demand for developing new applications whilst also facing an increasing shortage of IT professionals to meet this need. In response, they are adopting low-code development platforms and solutions provided not only by startups but also by tech giants such as Salesforce and Microsoft. The global low-code platform market was valued at almost $22.5 billion in 2022 and is expected to grow to $32 billion in 2024.[7] Based on a review of the available literature on low-code software development approaches, Table 1 summarizes the key characteristics, benefits and limitations of low-code development.[8]

In essence, the literature suggests that low-code development platforms can aid software development efforts in contexts where the offered functionalities of the platform match basic needs in simple work processes. However, the literature also indicates that using low-code development platforms might be more challenging for applications with complex functionality or a higher need for customization, let alone for the development of systems that integrate data science models. Thus, it remains unclear whether low-code development platforms are suitable for firms that seek to adopt or develop AI solutions. To better understand that conundrum, we next examine the potential challenges inherent to AI development.

The Complexities of AI Development

AI development requires a different approach than that used for traditional software development projects. For example, expert knowledge, such as input from data scientists, machine learning engineers, front-end engineers and production engineers, needs to be combined to build not only a learning model but also the application that is based on the model, meaning that AI development provides a particularly complex coordination challenge.[9] Moreover, AI development not only requires creating the front-end, back-end and user interface but also additional procedures, such as data preprocessing, model creation, training, evaluation, deployment and monitoring.[10] Once deployed, AI also needs to be monitored to manage and mitigate deployment issues that may arise, such as biases and drift.

To address these challenges, firms have begun to look for guidance on how to approach AI development. For example, Machine Learning Operations (MLOps) is a reference framework for developing and deploying AI models in production that builds on automating the AI development lifecycle end-to-end.[11] MLOps extends traditional DevOps approaches with AI-specific technical process steps, such as continuous training, to reduce the time needed for an AI software release, and to bind AI development and operations together more closely in a continuous pipeline.

Yet even when supported by MLOps, AI development remains a complicated and time-intensive process. For example, AI models must be part of a broader software ecosystem that hosts and provides access to AI model features as well as the constantly changing data that feeds

[7] Vailshery, L. S. *Low-Code Development Platform Market Revenue Worldwide 2018 to 2024*, Statista, February 21, 2024, available at https://www.statista.com/statistics/1226179/low-code-development-platform-market-revenue-global/.

[8] Alamin, M. A. A., Uddin, G, Malakar, S. and Afroz, S., op. cit., January 2023.

[9] van den Broek, E., Sergeeva, A. and Huysmann, M., op. cit., September 2022.

[10] Symeonidis, G., Nerantzis, E., Kazakis, A. and Papakostas, G. "MLOps—Definitions, Tools and Challenges," *Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, January 2022.

[11] Makinen, S., Laaksonen, E., Mikkonen, T. and Skogstrom, H. "Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?" *Proceedings of the 2021 IEEE/ACM 1st Workshop on AI Engineering—Software Engineering for AI (WAIN)*, May 2021.

Table 1: Characteristics, Benefits and Limitations of Low-Code Software Development

Characteristics	Benefits	Limitations
Visual programming: Instead of relying on textually created programming code in the development of software applications, low-code development tools map traditional software development processes to a graphical interface, on which users can use drag-and-drop techniques to build applications. **Abstraction:** low-code development tools use graphical abstraction for workflow, rules and data specification, and provide developers with a library of software components.	**Lowering skill barriers:** Little to no programming skills are required for using low-code development tools. **Outcome improvements:** Low-code development tools allow firms to better adapt to fast-changing requirements, develop applications that are closer to customers' needs and free IT resources for more complex tasks. **Complexity reduction:** Low-code development tools enable efficient collaboration between stakeholders and thereby reduce complexity, not only when developing new applications but also for ongoing maintenance and adaptability. They also allow firms to re-organize their software development processes and improve innovative behavior of employees across teams and functions. **Efficiency gains:** By reducing cost- and time-intensive revisions between IT and business professionals, low-code development tools can increase software development and delivery efficiency by a factor of five to ten. And because the tools are centrally provided software hosted by third-party providers, they also enable immediate prototyping of new requirements or testing of new functionalities with minimal effort.	**Adoption issues:** Many firms struggle to find the right low-code development platform for their needs and, in turn, waste time and resources due to a lack of knowledge about different platforms and how to use them. **Integration complexity:** Firms struggle with grafting low-code development platforms onto their existing IT infrastructures. **Technical limits:** Known issues include data and technology security concerns, reliability in providing solid and error-free service, application scalability in view of complex, large-scale and more essential applications, and the lack of flexibility that stems from the reliance on fixed building blocks in the software development process.

these models, which means that AI models need to be retrained even after release.

Because of this complexity, firms are looking to simplify AI development even further. For example, the AutoML approach, a cloud-based service, has been developed to completely automate the different process steps required for developing an AI model.[12] AutoML simplifies model creation, selection and the required data work considerably, and thus enables end users to quickly build an AI model without having to deep-dive into the complexity of model development or the lifecycle-dependent operations behind it. But AutoML also creates limitations. For example, though it eases the process of retraining, it requires additional computational power, is less flexible and is less suitable for specialized tasks.

In addition to AutoML, other low-code AI development platforms are becoming available to enable the easier development and deployment of AI applications[13] and have the potential to democratize AI in organizational settings. In addition to AI experts like data scientists and machine learning engineers, low-code approaches allow a broader range of users to be involved in the development of AI-based systems. Especially for smaller companies, which may not have the resources to explore how and where to make use of AI, a low-code approach to AI development could be a viable approach.

[12] Symeonidis, G., Nerantzis, E., Kazakis, A. and Papakostas, G., op. cit., January 2022.

[13] Prominent examples include the Bot Framework Composer, Botpress, Dialogflow and Kore.ai.

Table 2: Complementarities Between Low-Code Development and AI Development

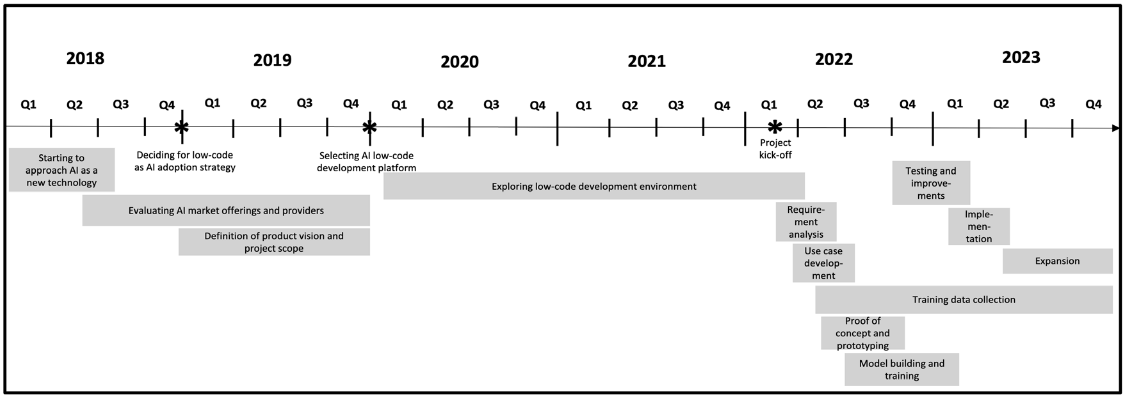
Characteristics of Low-Code Development Platforms	Characteristics of AI Development	Possible Complementary Capabilities of Low-Code Platforms for AI Development
Reduces development complexity	Poses a particularly complex development challenge	Low-code development platforms could be particularly effective in reducing the complexity of AI development by enhancing cross-functional collaboration between developers, business units and other stakeholders. Their accessible interfaces allow non-IT professionals to follow, understand and participate in the AI development process throughout all stages; all relevant stakeholders are enabled to take responsibility for and drive key processes themselves.
Simplifies software development steps	Contains additional development steps	Low-code development platforms could simplify and speed up AI development because all steps can be simplified for users. By drawing on the customization of pre-trained models while partially automating specific steps of AI development, they provide a promising option for AI-specific development steps, such as data preprocessing, model training and evaluation.
Accessible to unskilled users	Requires particularly skilled experts	Low-code development platforms promise to make software development skill requirements less critical, maybe obsolete. In this sense, a low-code approach to AI development could allow even resource-constrained firms to explore and make use of AI as much of the required functionality is provided through “drag-and-drop” building blocks in the low-code development platform itself. The model training and retraining processes, as well as the related data work, are partially automated, thereby reducing the need to involve AI experts such as data scientists.
Model-driven	Data-driven	Instead of requiring large amounts of input data, low-code development leverages drag-and-drop visual techniques to compose software out of ready-to-use features and building blocks. Thus, low-code AI development platforms can adapt pre-trained models and significantly reduce the amount of data that is required to develop an AI solution.

Complementary Characteristics of Low-Code Development Platforms and AI Development

Table 2 shows how the characteristics of low-code development platforms might complement the characteristics of AI development. Nevertheless, though the complementary capabilities of low-code development platforms for AI development appear promising, platform limitations also force organizations to carefully evaluate the trade-offs between the enablers and constraints. On the one hand, low-code AI development platforms might indeed enable the democratization of AI development. On the other hand, they could also constrain firms in their AI development.

To help organizations make the trade-offs between platform enablers and constraints, we carried out an in-depth case study of an AI development project based on a low-code development platform.

Figure 1: Timeline of GuideCom's Development of an AI-Based Smart Assistant


2018
2019
2020
2021
2022
2023
Q1
Q2
Q3
Q4
Starting to approach AI as a new technology
Deciding for low-code as AI adoption strategy
Selecting AI low-code development platform
Project kick-off
Evaluating AI market offerings and providers
Definition of product vision and project scope
Exploring low-code development environment
Require-ment analysis
Use case develop-ment
Training data collection
Proof of concept and prototyping
Model building and training
Testing and improve-ments
Imple-men-tation
Expansion

How GuideCom Leveraged Cognigy.AI's Low-Code Development Platform to Develop Conversational AI

GuideCom is a medium-sized German software development, consulting and change management company that specializes in the provision of HR management and banking software. GuideCom employs 230 staff who serve more than 600 client firms and 700,000 users.

In 2018, GuideCom embarked on a three-step strategic innovation to integrate conversational AI technology for end-to-end automated HR management services into its solution portfolio. The first step was to create a *smart assistant* that would always be available to answer questions that client employees might have relating to their employment. The second step was to build a *smart organizer* that would be able to proactively take over specific tasks for the employees. The third step was to provide a *smart coach* that would be able to perform leadership functions and support employees in skill and knowledge development.

We focus on how GuideCom implemented the first step of this strategic plan—the development of a smart assistant chatbot based on conversational AI technology. The timeline of the main stages of the project are depicted in Figure 1. The smart assistant was envisioned to provide HR services to client employees, which hitherto had typically been provided by HR managers. For example, employees could approach the smart assistant with questions concerning their employment contract, use it to report sick leave or apply for (parental) leave. These and similar HR service processes would be processed end-to-end by the smart assistant, meaning that it needed to be capable of processing all steps, from inquiry to application and approval within the system.

GuideCom's vision for the smart assistant product was that it would become a key module in the company's offerings that could be sold to all its current and future clients. To get started on this grand ambition, GuideCom developed the initial version of the smart assistant product in collaboration with one of its existing and most trusted clients, a major German bank. This bank was willing to be the pilot client and provided knowledge and feedback throughout the whole process from conceptualization to implementation, thus performing the role of innovation partner.

Because GuideCom had considerable software development expertise at the start of the project but no experience in AI development, it decided to use Cognigy.AI, a leading enterprise platform for low-code conversational AI development.[14] Below, we describe the four phases through which the development project unfolded and how

[14] For information about Cognigy and its AI solutions, see *Global Leader in AI-driven Customer Service Solutions*, Cognigy, available at https://www.cognigy.com/about.

Table 3: Key Stakeholders in GuideCom’s AI Development Project

Stakeholder	Role	Background
1	Head of Digital Transformation and chief digital officer	Responsible for the digital transformation of GuideCom and its core business. He is involved in the strategic development of GuideCom as well as in the actual practical implementation of the distinct projects. He devised the firm’s AI adoption strategy and managed the subsequent search and selection of a suitable technology partner.
2	Project lead	Responsible for evaluating market trends and fostering company-wide innovations, and for innovation management at GuideCom. Her tasks include innovation planning and management, as well as the complete portfolio planning related to GuideCom's software solutions. She leads the AI project, with the project team comprising system analysts, software developers, designers and a partnering customer.
3	Use experience designer	Responsible for the entire user experience at GuideCom, she evaluates existing as well as new applications under development from the users’ perspective in order to design an intuitive application that is easily useable.
4	System analyst	With a background in software development, he co-leads the AI project, representing the link between technical development and conception and design. His main task is to mediate between the product vision and its technical feasibility, thereby coordinating the independently working software developers and the rest of the innovation team.
5	System analyst (trainee)	Supports the system analyst as well as the project lead. His main responsibilities are the development of an appropriate interaction and designing the FAQ section of the smart assistant.
6	Software developer	Primarily responsible for all issues concerning one of GuideCom’s standard product offerings, particularly integration and networking of third-party systems. He is also responsible for these aspects of the AI project.
7	Software developer (trainee)	Involved in different software development projects at GuideCom. He was temporarily assigned to the AI project.
8	Software developer (trainee)	Also involved in different software development projects of GuideCom. He was temporarily assigned to the AI project.
9	HR manager at client partner	HR manager at a major German bank, which performs the role of innovation partner in the AI project. He supports the project on a regular basis with his expert knowledge on HR service processes as well as providing specific domain expertise.
10	Customer success expert at Cognigy	A customer success expert at Cognigy, which provides the low-code AI development tool used by GuideCom. She offers expert knowledge on the platform and how it can be used and integrated effectively.
11	Service designer	As a professional organizational psychologist with a background in industrial engineering, his expertise lies in corporate innovation and strategy design. He provides strategic-level assistance to the AI project, specifically supporting the identification, clarification and subsequent reorganization of service processes that are planned to be processed via the new system under development.

GuideCom managed to make use of the Cognigy.AI platform to manage the complexity of AI development. Table 3 provides an overview of the 11 key stakeholders involved in the project, all of whom were interviewed for this study.

Phase 1. Exploring the Market for AI Technologies and Selecting a Platform Supplier

GuideCom began exploring conversational AI technology in 2018 to evaluate potential use cases for its clients. This first phase, the investigation of AI technology, was less about finding a solution for a well-defined problem than about exploring AI as a promising new technology. Because the market already provided know-how concerning AI development, GuideCom researched the offers of different technology suppliers—from startups providing solutions for automating AI development process steps to big tech firms providing prefabricated AI models for specific solutions. For example, in addition to Cognigy.AI, GuideCom considered Microsoft's Bot Framework, AI21 labs and Amazon Web Service offerings. Its assessment of offerings focused on two aspects: 1) How does conversational AI technology work and what requirements does it impose for development, implementation and use? 2) How do different solutions for supporting the task of developing an AI-based system work and what would a cooperation with a technology provider look like?

GuideCom learned two important things from this exercise. First, to sustainably benefit from AI as a new technology, GuideCom needed to retain the ability to make its own decisions about what to do rather than being influenced by existing AI providers, including suppliers of prefabricated off-the-shelf AI solutions. As a consequence, GuideCom decided not to rely on standard solutions. The Head of Digital Transformation explained that the objective for GuideCom was to identify an offering that could be integrated into its modular solution suite and grow alongside the company's solution portfolio:

> *"[We looked at] smaller startups [and] larger start-ups but also [at] established companies, including the big tech companies [that] offer something in this [space]. And … we also exchanged ideas with IT service providers [that] … offer consulting and implementation support. … We also decided that we had to understand the technology better if we wanted it to flow into our products and offer us greater potential. … Just doing a small project and letting something develop, that's not going to be enough for us."* Head of Digital Transformation, GuideCom

The second lesson was about the overall complexity involved in AI development. This complexity, the Head of Digital Transformation explained, was the reason that GuideCom narrowed down the decision to a low-code approach to AI development:

> *"So, we looked at the technology a bit more intensively at the beginning and also talked to one or two startups that [had] built their own [natural language understanding] engine, vector databases and so on. … That's certainly interesting…. but you … [also] have to think about what competencies you need. … We … decided relatively clearly … [not] to start from scratch at the lowest level and say: we build everything … ourselves."* Head of Digital Transformation, GuideCom

Because of these two lessons, GuideCom decided to focus on simplifying the process of AI development by reducing complexity and preserving resources and identifying technology providers that would automate the steps of the development process. At the end of 2019, after the long-term product vision (starting with a smart assistant, then a smart organizer and finally a smart coach) had been developed, GuideCom screened providers that could provide technical platform support for the project. The criteria used for selecting an AI provider included delivery performance, possibilities for partnership, user experience design maturity, documentation, support for testing, integration capability (specifically programming languages, frameworks and APIs), technical administration, multilingualism, licensing model, costs, data security and governance.

GuideCom decided to partner with Cognigy, a leading provider that offers a low-code development platform for building an AI-based conversational agent. The Cognigy.AI platform automates the steps of model choice and building while also simplifying model training and retraining. The main reason for partnering with Cognigy was that this provider both offered a solid and trusted service and would interact with GuideCom at a peer-partnership level. This meant that Cognigy would not only act as an innovation leader for GuideCom but also had the expertise, manpower and consulting capabilities to support GuideCom through the development and management of the AI-based system. Moreover, GuideCom not only had faith in Cognigy's current product and services but trusted that it would stay abreast of further developments in the field—such as generative AI—meaning that GuideCom could rely on Cognigy's platform in the long-term well and beyond the scope of the smart assistant project.

Phase 2. Exploring the Low-Code AI Development Environment and its Enablers and Constraints

In the second phase, the business unit responsible for GuideCom's HR management suite took on responsibility for the project, focusing on product development and management processes. First, the business unit immersed itself in exploring the Cognigy.AI low-code development tool to more deeply understand the chosen platform. By studying the detailed documentation and taking advantage of Cognigy's training and consulting services, the business unit gradually became familiar with the functionality of the low-code AI development platform. This approach meant that Cognigy became an important part of GuideCom's knowledge network, which provided significant benefits for GuideCom during the AI development project. A software developer highlighted the partner role played by Cognigy, which enabled GuideCom's project team to consult with its partner on problems that needed to be solved:

> *"We were always in regular contact with one … consultant from [Cognigy] … who … answered various questions … about the general structure: … How do we … structure the FAQs? … How do we build these [conversation] flows? … He always gave [precise] tips on development-related questions about how we might be able to implement or solve certain things. … It was just special in that sense because we had never worked with [the Cognigy.AI platform] before and there were new concepts that we had to [learn about]."* Software Developer, GuideCom

Because the low-code AI development platform provided an accessible visual interface, GuideCom's IT professionals felt empowered to discover and test the platform iteratively and incrementally on their own. This allowed them to learn at their own pace, not only about the specific functionality of the platform but also about AI technology itself. One software developer described the intuitive nature of the Cognigy.AI platform as being helpful in exploring AI development:

> *"[We] basically stood there with the project team and had access to [the Cognigy.AI platform], where we could … try out various things. … Yes, we worked on it a bit and then more or less discovered and tried out the whole thing ourselves."* Software Developer, GuideCom

These learning opportunities became an early key enabler of AI development at GuideCom. However, it also became evident that architectural issues concerning the integration of the Cognigy.AI platform into GuideCom's existing HR solution suite would be a key challenge during the development of the smart assistant. A system analyst explained that overcoming the integration challenge would be an issue:

> *"You have a dependency on an external system in the development. You can't just install the [Cognigy.AI platform] on a local computer."* System Analyst, GuideCom

Three specific architectural issues emerged during the explorative learning process, which are summarized in Table 4.

The first architectural issue was related to security. The low-code AI development platform had to be able to ensure the safe use of the AI chatbot application, especially in terms of data provenance. The second issue related to integration. The AI chatbot application and its distinct components needed to be integrated with GuideCom's existing HR software suite. The third issue concerned expandability. GuideCom wanted

Table 4: Architectural Issues in Low-Code AI Development

Issues	Description
Security	GuideCom required the flexibility to implement additional, domain-specific security measures to complement the robust security features already provided by the low-code development platform.
Integration	GuideCom needed to build an application around the pre-trained AI model. Integration became an issue because of the decision to host the AI model on GuideCom's infrastructure.
Expandability	GuideCom needed its AI-based smart assistant to be expandable not only to different use cases—such as HR services or HR training and development—but also to different client firms. To achieve this, AI model building and training had to be automated from within GuideCom's HR software suite.

the AI chatbot application to be expandable to different firm clients.

The Cognigy.AI low-code AI development platform is a cloud-based service but GuideCom wanted to retain training and operational AI data on its own hosted infrastructure to ease maintenance and mitigate risks in the further expansion of the smart assistant. A system analyst emphasized the importance of remaining autonomous from the low-code platform:

> *"It is important to us that we have the data in [the HR software]. That is, we have the FAQs ..., how [work- and conversation flows] are connected ... ; everything is with us."* System Analyst, GuideCom

Phase 3. Identifying Use Cases and Developing the AI Model

Based on the insights gained during the first two phases, GuideCom then started building the first prototype of the smart assistant. In doing so, the project team recognized the need to deepen its understanding not only of AI technology but also of the HR processes that the smart assistant would eventually be deployed in. Thus, the team began to collect and analyze requirements.

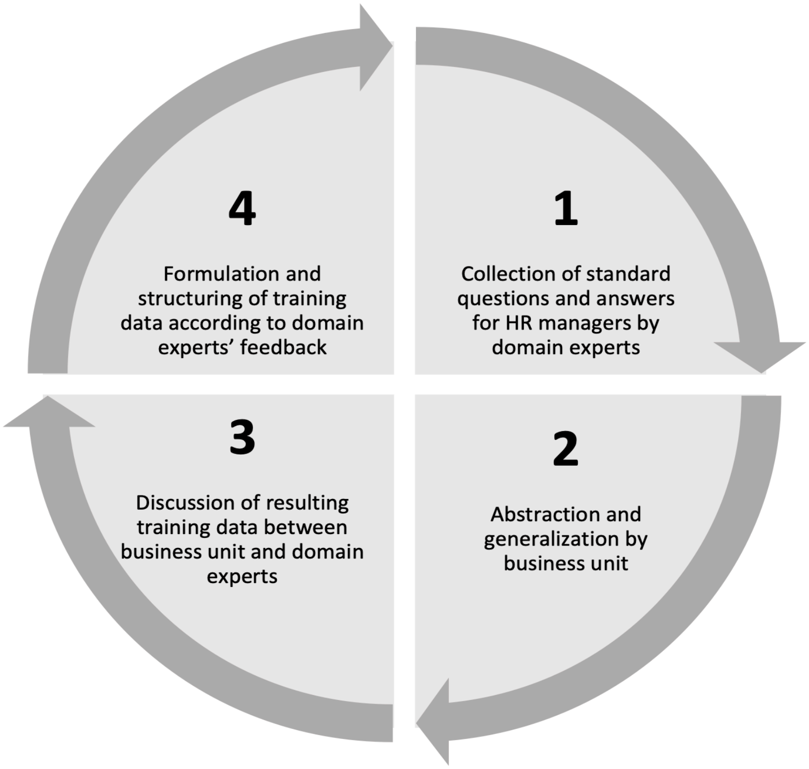
The vision was that the smart assistant would be a constantly available co-worker that delivers end-to-end automated HR services. To realize this vision, GuideCom's business unit first had to learn what the typical processes in an HR department were and how they unfolded. To achieve this, the team needed to collaborate closely with domain experts from an HR department. GuideCom therefore decided in the spring of 2022 to include one of its closest client firms, a major German bank, in the project as its co-innovation partner. The bank would play two major roles: identifying and developing use cases and collecting the data needed to train the AI application.

Both use case development and training data collection were completed manually and iteratively: The bank's domain experts collected typical questions and answers from the HR department, passing this data to GuideCom's business unit to format it according to the requirements of the platform. Because many questions were highly interdependent and not always immediately comprehensible to outsiders, constant consultations between the business unit and domain experts were necessary to develop use cases and sensibly train the envisioned AI model. Figure 2 provides an overview of the process steps that GuideCom and the bank conducted in tandem to collect and prepare AI training data.

The Cognigy.AI platform helped to enable this process. The visual interface of the platform facilitated the required cross-functional collaboration between IT and domain experts, which eased the interaction between GuideCom and the domain experts in the bank. The platform facilitated understanding and iterative development, and also guaranteed buy-in from the domain experts, who learned what kind and quality of data they needed to provide.

The platform also enabled the business unit to quickly build a prototype of the smart assistant, which allowed GuideCom to show its innovation partner (the bank) how it envisioned the evolution of the smart assistant product

Figure 2: GuideCom's Process for Collecting and Preparing Training Data for its AI Model


1
Collection of standard questions and answers for HR managers by domain experts
2
Abstraction and generalization by business unit
3
Discussion of resulting training data between business unit and domain experts
4
Formulation and structuring of training data according to domain experts' feedback

right from the start and to adapt it incrementally based on the bank's feedback. In turn, this helped the bank better understand what information it needed to provide and motivated it to engage in the time-intensive processes of use case development and training data collection. One of the bank's domain experts explained how the Cognigy.AI platform allowed it to see the tangible, evolving prototype, which motivated the bank to get involved in the development process:

> *"that was when we knew for the first time: where does the journey actually go and what do we get involved in now. ... We ... were very, very euphoric. .... From then on, it was really fun, because we knew ... what [we] are ... working on ... and what [it] will ... look like in the end. That was a really great day and a milestone for us as a whole in retrospect. We then thought: yes, that's probably how it can work. And then we really put our backs into it."* Domain Expert, client partner

GuideCom also realized that the low-code AI development approach required it to address impending issues relating to third-party control. Because the smart assistant was envisioned as a product to be licensed to many different clients, GuideCom needed to build a solution that would manage the dynamic and constantly evolving nature of AI[15] and handle changing data to fine-tune the AI model to the specific context of different client firms.

However, the workload for constantly reevaluating an AI model and retraining it is high, and firms like GuideCom typically have limited resources. GuideCom therefore decided to involve

[15] Berente, N., Gu, B., Recker, J. and Santhanam, R. "Managing Artificial Intelligence," *MIS Quarterly* (45:3), September 2021, pp. 1433-1450.

its innovation partner (the bank) in the process of model retraining for use cases other than its own. To enable this, GuideCom rebuilt and integrated relevant components of the Cognigy.AI low-code development platform so that it could add, adapt and refine training data (GuideCom's so-called intent trainer and FAQ manager) directly into its own HR software suite. Doing so allowed GuideCom's clients to retrain the AI model on their own via the Cognigy.AI platform's easy-to-access interface directly through GuideCom's HR software application.

A system analyst explained how the integration of the platform's components and the new smart assistant into GuideCom's existing IT architecture enabled client domain experts to retrain different AI models independently without being forced to understand and access the platform itself:

> *"for the user, everything is ultimately there in [the HR software application]—that is, all the perspectives [view and manage FAQ, intent trainer and chat component] and also the chatbot. It's just that [in the background], a lot of things happen on [the Cognigy.AI platform.]"* System Analyst, GuideCom

By adopting this approach, GuideCom learned that customization, expandability and integration of third-party knowledge would be difficult to achieve without additional software development expertise. The firm realized that even low-code AI development required foundational programming knowledge, especially when requirements emerge for the AI-based system that deviate from the default pathways that a low-code development platform provides. However, the Cognigy.AI platform eased the process of AI development by empowering non-programming users to contribute to product development through the easily accessible interface. For instance, the developers found that building and training the AI model was not a key challenge. The platform automated the central steps, such as choosing the best fitting model for the given purpose, acquiring, preparing and pre-processing large data sets to train the model for the basic understanding of language, deciding which framework to build the model in, defining its architecture and testing its performance.

Other key steps, such as customizing the model with domain-specific data through training and retraining, were not automated but simplified through the Cognigy.AI platform's visual interface. The platform features enabled AI model developers to focus mainly on building workflows on the platform, and defining which information was supposed to be communicated in which way for different intents, and which service processes had to be triggered in response. To illustrate, Figure 3 provides a screenshot of Cognigy.AI's interface.[16]

A developer described the different functionalities the platform stated:

> *"[the Cognigy.AI platform offers] a lot of ... expertise ... [it] takes a lot of work off your hands [especially] preparatory work ... you don't just build Cognigy.AI like that. So, I think that [the platform is] an immense advantage [for getting] into the subject [of AI development]."* Software Developer, GuideCom

In summary, the Cognigy.AI low-code AI development platform reduced the complexity for GuideCom's business unit in developing the AI-based application, allowing the business unit to focus on data collection and preparation for AI-model training, as well as addressing security, integration and expandability issues.

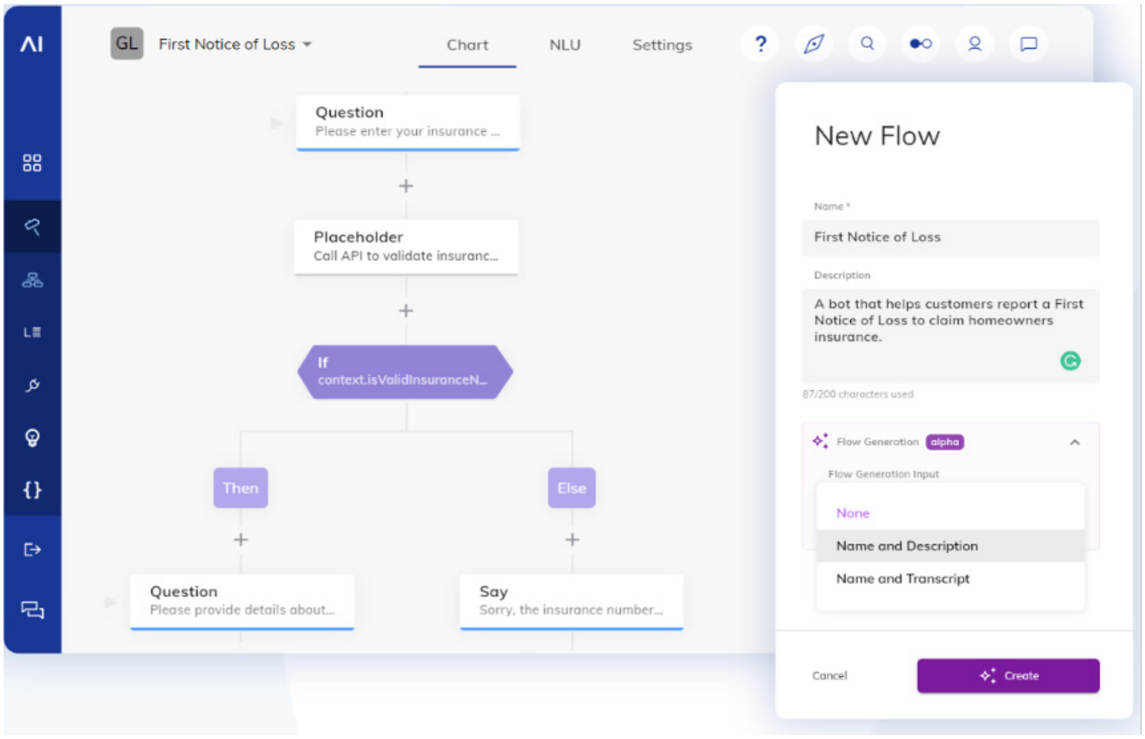
Phase 4. Testing, Improving and Implementing the Final Product

To prepare for the implementation of the final smart assistant product, GuideCom began collecting user feedback to test and improve it. Drawing on the innovation partner's network, the business unit used both qualitative and quantitative testing techniques to evaluate how users were experiencing the AI-based smart assistant. Table 5 provides an overview of the 13 users who tested and evaluated the AI-based system, and the feedback they provided.

A service designer explained that the purpose of user testing was to evaluate both experience and functionality:

[16] Source: *Enterprise-Grade Contact Center AI Platform for Customer Service Transformation*, Cognigy, available at https://www.cognigy.com/platform/cognigy-ai.

Figure 3: Screenshot of Cognigy.AI's Visual Programming Interface



New Flow

Name *

First Notice of Loss

Description

A bot that helps customers report a First Notice of Loss to claim homeowners insurance.

87/200 characters used

> *"[The purpose of user testing was] to make sure that the features of the smart assistant work [and] are easy to use from the user's point of view ... so basically, the whole criteria around user experience and usability. And I would distinguish [between features and usability.[The former refers to] testing ... [that the smart assistant basically works], is error-free and so on. And [the latter refers to the] user experience, [checking that] it is ... fun and purposeful to work with."* Service Designer, GuideCom

Based on the testing feedback, GuideCom developed a plan for what and how to improve before or after the implementation of the smart assistant, depending on the urgency of the changes. Adjustments that needed to be made before implementation included the visual and interface design of the FAQ section.

In this final project phase, GuideCom again benefited from its existing software development experience, which allowed it to customize the final product and fix many of the front-end-related design issues through direct code re-engineering. The user tests also identified a problem with the AI model the smart assistant was based on, specifically with the intent recognition process, which suggested that the training process needed to be intensified with more and diverse data on the different intents. But because the cooperation with the client firm was already well established and the low-code environment easily accessible, these changes were straightforward to implement.

While these adjustments were being made, the business unit started to prepare to implement the smart assistant. The preparation included a presentation and a targeted marketing campaign, informing employees at the partner bank about the functions and new HR processes the smart assistant would provide. The marketing campaign included an internal website at the bank, which introduced and explained the new AI-based assistant and its functions. GuideCom also organized internal training sessions in

Table 5: Test Users and Excerpts from Test Feedback

Test User	Background	Key Feedback
1	HR manager; provides HR services at the innovation partner's firm	Initially not clear on how to enter and interact with the system; answers were partly too long and too detailed; unclear about data security; hand-over to the HR manager is important
2	Manager of digitalization of "People & Culture" at the innovation partner's firm	Design (interface, fonts) not perfectly readable; the system should initially provide all information on a topic before it offers service processes; expected a disclaimer on data security
3	Team coach of HR software suite at GuideCom	Feels impersonal compared to interactions with the HR manager; the ability to start and process services directly from the system is convenient; efficient; partly provides too much and unspecific information
4	Team coach of HR software suite at GuideCom	Needs a broader knowledge base; intent recognition and keyword search needs to be improved; knowledge that is already retrievable is good
5	Transformation manager at the innovation partner's firm	Comprehensible answers; introduction to the systems is important; prefers full-screen interface; hand-over to HR manager is expected
6	Employee experience manager at GuideCom	Expects tutorial on first use; content with answers; intuitive usage; structure of topics and intents should be improved
7	Business development manager at GuideCom	Answers partly too long; appreciates chat function; information given is helpful
8	Digital human resources at GuideCom	Intuitive usage; helpful answers; design of follow-up links should be improved; quality of information is important
9	Business and organizational development at GuideCom	Structuring of topics and intents should be in a procedural manner; answers are too generic
10	Branch manager at the innovation partner's firm	Practical, efficient, fast processing of service inquiries; introduction to the system would be helpful
11	Customer service at the innovation partner's firm	Initially not clear on how to enter and interact with the system; intent recognition needs to be improved; mostly content with the answers
12	Branch assistant at the innovation partner's firm	Needs more knowledge to act as an assistant; answers partly too long for interface format
13	Assistance in the IT department at the innovation partner's firm	At first, not clear how to enter and interact with the system; content with answers; appreciates human-like communication style

cooperation with the bank to help the bank's employees understand and efficiently work with the new system. The project lead highlighted how important this ongoing communication and support were to the success of the implementation process:

> *"We … have different formats … [for] informing the employees about how the whole thing works, and how [the smart assistant] comes to an answer, … and where the questions and answers come from, [so] that the customer is also very much involved. … I think this … transparency is also important."* Project Lead, GuideCom

Lessons Learned that Underpin a Framework for Managing Low-Code AI Development

To capture the strategic lessons from the GuideCom case we have encapsulated them in the conceptual framework shown in

Figure 4: Enablers and Constraints of Low-Code AI Development

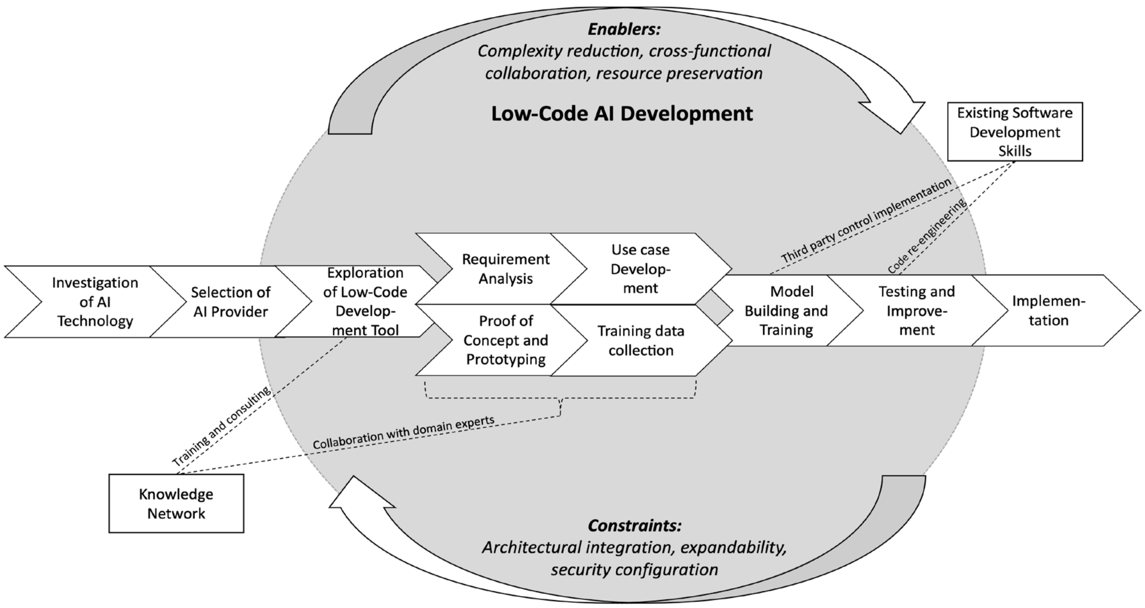

Enablers:
Complexity reduction, cross-functional collaboration, resource preservation
Low-Code AI Development
Existing Software Development Skills
Third party control implementation
Code re-engineering
Investigation of AI Technology
Selection of AI Provider
Exploration of Low-Code Develop-ment Tool
Requirement Analysis
Use case Develop-ment
Proof of Concept and Prototyping
Training data collection
Model Building and Training
Testing and Improve-ment
Implemen-tation
Training and consulting
Collaboration with domain experts
Knowledge Network
Constraints:
Architectural integration, expandability, security configuration

Figure 4. Companies looking to use low-code AI development platforms to build AI-based products can use this framework to guide their endeavors.

The framework visualizes a low-code development approach to AI development as a process that requires making trade-offs between enablers and constraints that stem from the choice of platform. The conflicts between these two elements of a low-code platform need to be resolved throughout the process that is situated in contextual factors (in particular, the existence of knowledge networks and software development skills) that influence the extent to which low-code development platforms can be successfully leveraged in the process of exploring, prototyping, building, testing and implementing an AI product.

Below, we discuss the three lessons learned that underpin our framework.

Lesson 1. Playful Exploration and Use-Case Centricity are Key to

AI Development with Low-Code Development Platforms

Our framework shows the distinct steps involved in the process of developing an AI-based product with the help of a low-code development platform. Most companies begin their AI journey by recognizing that AI is a rapidly evolving technology and set about investigating possible ways of adopting AI for business solutions. Because of the growing interest in the low-code development approach, which is a promising prospect, especially for small and medium-sized companies with limited experience in AI development and resources, the next step is to select a low-code platform provider able to meet the company's needs. As our case shows, low-code AI development platforms provide accessible visual programming interfaces, which lower the barriers to exploring and learning substantially, allowing the users to "playfully" start building trial concepts immediately.[17,18]

The actual development of an AI-based application using a low-code approach is then driven by an intensive collaboration between IT and domain experts, facilitated through the easy-to-access interface of a low-code AI development platform. Jointly developing specific use cases to orient the AI development process means domain experts are empowered to focus on and drive the data work (collecting and preparing the data needed to train the AI model).[19] The joint development of use cases before starting the actual AI development will be easier if the collaborators proactively explore and familiarize themselves with the low-code AI development platform in advance. To do this, they should embark on a phase of "playful exploration," where trial concepts are built, enabling both IT and domain experts to gain an understanding of the tasks, skills and roles needed for low-code AI development and to subsequently divide responsibilities accordingly.

To prepare for AI model building and training and its integration into the existing software architecture, companies typically go through a phase of prototyping and collecting training data. In this phase, they foster collaboration between the IT department and domain experts. They should also reflect on their needs and expectations for the further development, retraining and maintenance of the AI-based application in the making. Specifically, they must determine the extent to which third-party control is required. For example, if an AI solution must be expandable (across different departments, use contexts or firms), they must allow third parties—either domain experts from different fields or from other firms—to retrain the model in their context and with their data. This is a challenge because AI models are particularly sensitive to use cases and context.[20] However, third-party control is not necessarily required if an AI-based system is built exclusively for one use case, because training and retraining could be undertaken centrally, for example by the IT department.

Finally, there will be a phase of testing and improvement before the actual implementation of the AI-based product. The GuideCom case shows that testing and improvement—though still relying on the low-code AI development platform—benefit from additional software development expertise. Experienced developers will more effectively complete reengineering of the developed AI code to achieve a customized and improved solution.

Lesson 2. Low-Code Development Platforms Both Enable and Constrain AI Development Processes

The framework in Figure 4 shows how AI low-code development platforms act as a three-pronged enabler to the overall AI development process. First, these platforms reduce the complexity of AI development by automating several of the process steps and hiding them behind an accessible visual interface. For example, the visual interface enables users to focus both on training and retraining an AI model and designing the application around it.

Second, low-code AI development platforms facilitate cross-functional collaboration between developers, business units and clients or customers. By allowing non-IT professionals to follow, understand and participate in the AI development process throughout all stages, all stakeholders can take responsibility for and drive key processes themselves. For example, domain experts are able to collect and prepare the data that is necessary to develop, expand and adapt the AI-based product, because they are guided by the low-code development tool on the form, quantity and quality of the data needed.[21]

Third, low-code AI development platforms can allow even resource-constrained firms to explore and make use of AI because much of

[17] Li, L. and Wu, Z. "How Can No/Low Code Platforms Help End-Users Develop ML Applications? A Systematic Review," in *HCI International 2022—Late Breaking Papers: Interacting with eXtended Reality and Artificial Intelligence (Proceedings of the 24th International Conference on Human-Computer Interaction, HCII 2022, Virtual Event, June 26 – July 1, 2022)*, pp. 338-356.

[18] Lins, S., Pandl, K. D., Teigeler, H. and Thiebes, S. "Artificial Intelligence as a Service," *Business & Information Systems Engineering* (63:4), August 2021, pp. 441-456.

[19] van Giffen B. and Ludwig, H. "How Siemens Democratized Artificial Intelligence," *MIS Quarterly Executive* (22:1), March 2023, pp. 1-21.

[20] Geske, I., Hoffman, P., Lämmermann, L. Schlatt, V. and Urbac, N. "Gateways to Artificial Intelligence: Developing a Taxonomy for AI Service Platforms," Proceedings of the 2021 *European Conference on Information Systems (ECIS)*, ECIS Research Papers 150, 2021.

[21] van Giffen B. and Ludwig, H., op. cit., March 2023.

the required functionality is provided as "drag and drop" building blocks in the platform itself. Processes of (re)training and related data work are streamlined by low-code AI development platforms, thereby reducing the need to involve AI experts such as data scientists.[22] Furthermore, because low-code development platforms enable the domain-specific adaptation of a pretrained AI model, they reduce the amount of data that is required substantially. In the GuideCom case, this meant that the firm could simply and manually collect, prepare and process all the training data that was needed.

In addition to these enabling factors, the framework in Figure 4 highlights that low-code AI development platforms can also constrain AI development. First, a platform—and the AI model it contains—is yet another system that, to become accessible and productive, needs to be integrated both into a software application and also into the existing IT infrastructure. The challenge of these integration issues is greater than traditional systems integration[23] because it is exacerbated in the context of AI development by the inherent "frame" problem.[24] This problem arises when low-code AI development platforms are used to develop a product that will be deployed for different clients with different infrastructures and different context, requiring the AI model to be (re)trained on different data. The same holds if an AI-based system will be used for different use cases.

The second constraint relates to expandability. An AI product that will be used by different clients or for different tasks will inevitably have to be trained multiple times with different data.[25] Thus, the software architecture the AI model is embedded in should allow for third-party control to safeguard the expandability of the AI-based product.

The third constraint placed on AI development by low-code development platforms relates to platform security. This may include issues like product safety, access management, the security of data that needs to be transferred to external servers or a cloud to train the externally built AI model, and user privacy. Organizations that use low-code AI development platforms must rely on safety measures related to cloud-based platform provision, which may be incomplete in terms of the security, privacy or confidentiality demands of third-party clients or regulatory and legal institutions. For instance, privacy policies of low-code AI development platforms may vary depending on where the provider is located. If companies want or need to comply with the General Data Protection Regulation, they must select a provider that complies with this legal framework. The GuideCom case shows that one way in which this constraint comes into play is when firms deploy externally built AI models, train them with internal data and then deploy them in the context of third-party clients.

Lesson 3. Software Development Skills and Knowledge Networks Are Still Needed for AI Development

The final aspect that the framework in Figure 4 highlights is that both the enablers and constraints of low-code AI development platforms apply within the specific contexts of the firms that use this approach. Though firm contexts can vary on several dimensions (such as resource base, regulatory pressures, market environment or digital orientation), two characteristics of this context are particularly relevant—the extent of existing software development skills and the possibility of accessing a broader knowledge network.

At GuideCom, the existing software development skills meant that it could benefit more easily and more readily from the enablers inherent in the low-code AI development platform.[26] For example, because GuideCom's leading business unit had already benefited from the complexity reduction of the low-code development platform, it was able to adapt the AI-based system to the unit's needs by re-engineering the platform's code and thus increase complexity where needed. Doing this meant it had to deviate from the given pathways the low-code development platform provided. This was of particular importance for the implementation

[22] Reilly, J. "How No-Code Platforms Can Bring AI to Small and Midsize Businesses," *Harvard Business Review*, November 5, 2021.

[23] Goodhue, D. L., Wybo, M. D. and Kirsch, L. T. "The Impact of Data Integration on the Costs and Benefits of Information Systems," *MIS Quarterly* (16:3), September 1992, pp. 293-311.

[24] Salovaara, A., Lyytinen, K. and Penttinen, E. "High Reliability in Digital Organizing: Mindlessness, the Frame Problem, and Digital Operations," *MIS Quarterly* (43:2), June 2019, pp. 555-578.

[25] Geske, I., Hoffman, P., Lämmermann, L. Schlatt, V. and Urbac, N., op. cit., 2021.

[26] Sundberg L. and Holmström, J., op. cit., 2023.

of third-party control and the development of customized solutions in the testing and improving phase of developing the AI-based product.

GuideCom also benefited from its knowledge network, consisting of the platform provider (Cognigy) and its innovation partner (the bank), which helped members of GuideCom's project team to familiarize themselves with the low-code environment and, later on, customize the AI-based product for different use cases and clients. For instance, Cognigy provided support, consultancy and training services, which helped GuideCom's developers and business unit to quickly and intensively interact with the low-code AI development platform, thereby eliminating knowledge barriers, enabling learning and speeding up the development process.[27] GuideCom also benefited from the close cooperation with its innovation partner, which constantly provided HR domain expertise and knowledge, and the data necessary to develop the AI-based chatbot tailored to its defined use cases.

Reflections on Lessons Learned

In absorbing the lessons learned from the GuideCom case, it is important to remember that GuideCom is a well-respected software development company. From the beginning, the firm's intention was to develop an AI-based chatbot system as a market offering rather than as an internal HR solution. In developing the AI-based product, it benefited both from its solid foundation in software development expertise and existing IT infrastructure and from its willingness to invest in AI development as a key product development capability. GuideCom recognized that investing in the development of the AI-based chatbot would ultimately provide a direct positive impact on its product portfolio and revenue potential.

Recommendations for Practitioners

From our analysis of the GuideCom case, we provide four recommendations for practitioners in small and medium-sized companies that are considering using low-code development platforms to develop AI-based products.

1. Acquire Software Development Knowledge Before Adopting the Low-Code AI Development Approach

Though the GuideCom case study shows the benefits of adopting a low-code approach to AI development, it also highlights that low-code does not necessarily mean that no coding experience is needed. Companies considering using low-code tools for AI development therefore need to critically evaluate their skills. This evaluation should consider the type of AI-based application that is planned, the specific use cases it is intended to serve, the target users and the knowledge and resources available to the company. The main questions companies should ask themselves are:

- Are we planning to build an AI-based application for one specific purpose only, or do we plan to expand it to different use contexts or even sell it as a product?
- How much customization does our AI-based application need, based on the purpose it is designed for?
- How many resources and knowledge on AI and software development do we have or can we access?

The GuideCom case shows that low-code approaches to AI development certainly offer a valuable middle way between resource-intensive in-house AI development and the costly and rather inflexible approach of buying an off-the-shelf AI solution. But adopters of the low-code approach should consciously weigh the benefits and challenges, as well as their capabilities and expectations.

2. Build Cross-Functional Collaboration Between IT and Business Domain Experts

The GuideCom case shows that the use of low-code AI development platforms empowers domain experts to take on meaningful tasks in the development process of AI-based products by fostering their understanding of the technology and the requirements of AI development. It also shows that business experts can proactively assume responsibility for the development of use cases. For each use case, they can then prioritize

[27] Krejci, D., Iho, S. and Missonier, S. "Innovating with Employees: An Exploratory Study of Idea Development on Low-Code Development Platforms," *Proceedings of the 2021 European Conference on Information Systems (ECIS), ECIS Research Papers 118, 2021.*

the collection and processing of the required training data, obtaining it from their day-to-day business activities, which ensures the AI-based system under development is customer-focused. Because low-code AI development platforms enable this cross-functional collaboration, they can facilitate the development of AI components without the need for developers and domain experts to deep-dive into AI-specific tasks and processes, such as data science or algorithm curation.

3. Choose a Low-Code AI Development Platform that Suits Your Needs for Integration and Expansion

In addition to clarifying, in advance, the AI use cases and the needed training data, firms adopting a low-code AI development approach must consider two particular requirements: integration needs and expansion needs. Integration issues relate to how an AI model can be embedded into a software application and how, in turn, this application can be embedded into the existing IT infrastructure. Firms should evaluate the integration options platform providers offer. In terms of expansion, firms should verify how expandable the AI model needs to be—i.e., will it need to be portable to different use cases or clients.[28] AI solutions are context-dependent, so expandability is typically complex or even limited.

Because firms will have to depend on safety measures provided by the chosen low-code AI development platform, they must also clarify whether prospective platforms are compatible with the architectural structures needed to support their (or potential clients') demands for data security and privacy. Most low-code development platforms are hosted on an external cloud and generally require their users to transfer the data needed to train the AI model to the cloud as well. Adopters of low-code platforms should thus be either able to permit this kind of data transfer or select a platform provider that offers on-premises hosting.

A key recommendation when choosing a low-code AI development platform is that firms should clarify, in advance, the strategic purpose of AI development and identify the key requirements that flow from this purpose. Goal requirements should then be mapped to selection criteria that help firms navigate the process of searching for and selecting suitable low-code AI providers. The criteria GuideCom used when selecting Cognigy.AI included delivery performance, possibilities for partnership, user experience design maturity, documentation, support for testing, integration capability (specifically programming languages, frameworks, and APIs), technical administration, multilingualism, licensing model, costs, data security and governance.

4. Establish a Knowledge Network

We found that access to third-party knowledge—in particular, regarding the low-code development platform—is a key success factor in low-code AI development. Companies looking to adopt a low-code AI development platform should therefore establish a knowledge network that provides expertise on both the low-code platform and the target application domain. We recommend that companies should only consider low-code development platform providers that offer training and consulting services. Furthermore, practitioners responsible for exploring ways to implement AI in small- and medium-sized firms—especially those with limited resources—should collaborate with innovation partners, who can contribute important domain expertise and introduce them to the low-code platform from the beginning of the relationship.

Concluding Comments

Our research found that, in terms of democratizing AI development, low-code AI development platforms are a mixed blessing. On the plus side, the enablers of these platforms substantially lower the barrier to AI development and shift skill and resource demands to selected stages of the AI development process. On the minus side, firms seeking to drive innovation with low-code AI development platforms must consider the constraints relating to integration, expandability and security, and find strategies for addressing them effectively.

[28] Huang, J., Henfridsson, O. and Liu, M. J. "Extending Digital Ventures Through Templating," *Information Systems Research* (33:1), August 2021, pp. 285-310.

Data Sources

Empirical Data									Total
Key informants	**Head of Digital Transformation**	**Project Lead**	**User Experience Designer**	**System Analysts**	**Software Developer**	**Client Domain Expert**	**Customer Success Expert**	**Service Designer**	
Interviews	1	2	2	3	3	1	1	2	15
Length (minutes)	53	132	117	157	84	64	33	121	761
Participant Observations									24 meetings
Field Notes									47 pages
Site Visits									3 days
Secondary Data									**Total**
Presentation slides, records from the online collaboration platform, internal project documentation									149 pages
Product and software demos									2

Appendix: Research Process

Between March 2022 and June 2023, we conducted 16 months of qualitative fieldwork, from the launch of GuideCom's smart assistant project to its final client implementation. We followed the operational phases of conception, development and implementation of the smart assistant from beginning to end in an exploratory manner, using an iterative process of observing the process, understanding the process and identifying central problems. The table above summarizes the sources of our data.

In total, we carried out 15 semi-structured interviews with all key stakeholders involved in the innovation project (Table 3). We also completed several hours of participant observations at internal and official meetings, design workshops, strategy planning appointments and user testing sessions. Additionally, we collected internal materials from the company, such as project documentation, product presentation slides and extracts from the internal platform as well as the online collaboration tool, which mapped all processes and the different branches of action, as well as related key decisions. We stored all data in a research database.

About the Authors

Imke Grashoff

Imke Grashoff (imke.grashoff@uni-hamburg.de) is a Ph.D. student at the University of Hamburg. Her research interests focus on the impact of digital technologies on organizations, collaboration and the future of work in general, and, in particular, the design, development and deployment of AI.

Jan Recker

Professor Dr. Jan Recker (jan.christof.recker@uni-hamburg.de) is an AIS Fellow, Alexander-von-Humboldt Fellow, Nucleus Professor for Information Systems and Digital Innovation at the University of Hamburg, and adjunct professor at the Queensland University of Technology Business School. His research focuses on digital innovation and entrepreneurship, digital solutions for sustainable development and systems analysis and design.