MIS Quarterly Executive

Volume 22 | Issue 3 Article 6

September 2023

Fueling Digital Transformation with Citizen Developers and Low-Code Development

Ainara Novales

Rubén Mancha

MIS QUARTERLY EXECUTIVE

Fueling Digital Transformation with Citizen Developers and Low-Code Development

As digital technologies reshape how companies create value propositions, organizational success and survivability will increasingly depend on continuous innovation and digital transformation. Gradually, organizations are adopting low-code development to fuel digital transformation and to increase responsiveness and customer orientation while counterbalancing a shortage of software developers. We describe the journeys of two early adopters of low-code development. Based on insights from these cases and interviews with executives from other organizations, we provide recommendations for adopting low-code development.[1,2]

Ainara Novales
Erasmus University Rotterdam
(the Netherlands) and
adesso SE (Germany)

Rubén Mancha
Babson College (U.S.)

Tackling the Development Bottleneck with Low-Code and Citizen Developers

An organization's ability to transform, fend off competition, and create and exploit business opportunities depends on its capacity to use digital technologies to continually enhance its offerings, processes, and business models. Digital technologies are changing both companies' processes and offerings. Paradoxically, though digital technologies create numerous opportunities for value creation, firms need to spend more time carefully designing their offerings, digital platforms and infrastructure,[3] which extends time to market. Companies must deal with the need for increased responsiveness and customer orientation provided by digital innovations while managing the shortage and high expense of software developers. To tackle the software development bottleneck while driving digital transformation, companies are increasingly turning to low-code development (LCD).

Low-code development platforms, such as Mendix, Microsoft Power Apps and OutSystems, are increasingly employed in consumer- and industrial-oriented businesses to power digital transformation and speed up development. Low-code development enables user developers, nontechnical employees and business partners to innovate and develop new software

[1] Blake Ives is the accepting senior editor for this article.

[2] The authors thank Blake Ives and the members of the review team for their thoughtful and constructive feedback and guidance during the review process. We also gratefully acknowledge the support of the interviewees from Hortilux, Volvo Group and seven other companies in sharing insights about the implementation of low-code in their organizations.

[3] In Yoo, Y., Boland, R. J., Lyytinen, K. and Majchrzak, A. "Organizing for Innovation in the Digitized World," *Organization Science* (23:5), September 2021, pp. 1298-1408. This phenomenon is referred to as the paradox of pace.

DOI: 10.17705/2msqe.00083

applications[4] that enable quick iterations in the development of digital solutions. Citizen development relying on low-code application development enables organizations to overcome the shortage of software developers and speed up their digital transformations.[5] Gartner predicts that the user base for low-code development outside formal IT departments will grow from 60% in 2021 to more than 80% by 2026.[6]

According to Nadella and Iansiti,[7] "Where [low-code] technology is architected to share critical capabilities, data, and information, it can empower a broad range of "professional" [software developers] and citizen developers to break down isolated silos and drive innovation across traditional boundaries."

We investigated two early adopters of LCD—Hortilux Schréder (Hortilux), a leading provider of grow light solutions for greenhouses, and Volvo Group,[8] a leading manufacturer of trucks, buses, construction equipment, and marine and industrial engines. We also interviewed nine executives at seven organizations drawn from diverse industries (for more details, see the Appendix). We sought to discover:

- How early adopters use low-code development to digitally transform their products, services and internal business processes
- Key actions that organizations need to take to successfully deploy low-code.

The implementation journeys of Hortilux and Volvo Group reveal how low-code tools democratize and boost agility in the development of digital offerings and the automation of processes. They do so by placing development tools in the hands of non-IT professionals who use them to enhance coordination and collaboration across heterogeneous players. Though the two case companies experienced different journeys and relied on different low-code platforms, our findings, drawn from these cases and interviews with executives in other companies, suggest that chief information officers (CIOs), chief digital officers (CDOs) and business leaders must:

1. Set a clear strategy for selecting low-code development use cases
2. Identify and assign low-code users and provide training to upskill/reskill tech-savvy employees
3. Establish a low-code team that provides organization-wide support
4. Ensure that the low-code architecture is up to date
5. Evaluate the technical requirements of the solutions to assess which low-code platform is the best fit.

Origin of Low-Code Development and Platforms

The term "low-code" describes a coding approach that is easy and simple to use and allows a wide range of users to develop software applications.[9] The roots of low-code development platforms can be traced back to the 1980s and 1990s with tools for computer-aided software engineering, rapid application development and business process management. However, the speed, agility and flexibility required for today's application development have in recent years fueled a wide adoption of LCD tools and platforms.[10]

Compared to traditional computer programming languages with complex coding syntax, LCD stays at a high level of abstraction and uses basic modeling principles. LCD facilitates the development of software applications and solutions using graphical user interfaces, declarative programming and drag-

[4] See: 1) Nadella, S. and Iansiti, M. Want a More Equitable Future? Empower Citizen Developers, WIRED, December 9, 2020; and 2) Johannessen, C. and Davenport, T. *When Low-Code/No-Code Development Works—and When It Doesn't*, Harvard Business Review Digital Article, June 22, 2021.

[5] Carroll, N. and Maher, M. "How Shell Fueled a Digital Transformation by Establishing DIY Software Development," *MIS Quarterly Executive* (22:2), June 2023, pp. 99-127.

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

[7] Nadella, S. and Iansiti, M., op. cit., December 9, 2020.

[8] We investigated Volvo Group, which is legally known as Aktiebolaget Volvo (shortened to AB Volvo). The car manufacturer Volvo Cars was part of AB Volvo until 1999, when it was sold to the Ford Motor Company.

[9] See Richardson, C. and Rymer, J. R. *New Development Platforms Emerge for Customer-Facing Applications*, Forrester Research, June 9, 2014, available at https://www.forrester.com/report/New-Development-Platforms-Emerge-For-CustomerFacing-Applications/RES113411

[10] Bloomberg, J. *The Low-Code/No-Code Movement: More Disruptive Than You Realize*, Forbes, July 20, 2017, available at: https://www.forbes.com/sites/jasonbloomberg/2017/07/20/the-low-codeno-code-movement-more-disruptive-than-you-realize.

Table 1: Main Differences Between Traditional and Low-Code-Based Application Development

	Traditional Application Development	LCD-based application development
Role of Non-IT Professionals	Information resources (contributor role)	Application designers (contributor role) and (co)developers (producer role)
Role of IT Professionals	Software developers	(Co)developers (supporting role and experts for quality assurance and architecture definition)
Software Development	Traditional hand-coded programming languages (e.g., C, C++, Python, Java)	Drag-and-drop development based on graphical user interfaces and declarative programming
Development Method/ Environment	Waterfall or Agile	Agile (actively involving business team members in the development)

and-drop development that connects triggers and actions. An LCD environment integrates visual modeling and coding programming functionalities. This simplifies professional developers' work, accelerates development, enables codevelopment efforts by mixed teams of technical and nontechnical employees and even allows non-IT citizen developers to create applications that optimize their own workflows. LCD has the potential to drive digital transformation by facilitating process automation and optimization, as well as promoting innovative methodologies that involve teams comprising both technical and nontechnical experts.

Though traditional and LCD-based application development share many characteristics (e.g., data storage systems, operational backbone), there are differentiating elements. LCD platforms are more intuitive and easier to use than traditional hand-coded programming languages, which are complex and require technical skills. Through using LCD's graphical interfaces and simple programming, nontechnical employees play a more central and active role in development and actively implement and test their ideas instead of passively sharing them.[11]

LCD platforms provide a more agile development environment, often involving other business team members, which dramatically and significantly accelerates application delivery[12] while ensuring that the delivered systems better meet the business requirements. Furthermore, artificial intelligence advances promise to bring intelligent capabilities to LCD and programming assistants to reduce the barriers to using LCD.[13]

Commercial LCD platforms, such as Appian, Mendix, Microsoft Power Apps, OutSystems and ServiceNow are rapidly spreading across companies and industries. These LCD platforms support a variety of use cases, including process automation, data integration, dashboarding and more advanced application development.

Two LCD Adoption Journeys

Below, we describe the low-code adoption journeys of our two early adopters: Hortilux and Volvo Group. Hortilux sought to transform its products by developing a comprehensive Internet of Things (IoT) platform based on the Mendix low-code tool. Volvo Group chose Microsoft Power Apps to empower citizen developers by reskilling and upskilling employees in business departments. For each case, we examine the reasons for adopting a low-code approach,

[11] Maruping, L. M. and Matook, S. "The Evolution of Software Development Orchestration: Current State and Agenda for Future Research," *European Journal of Information Systems* (29:5), September 2020, pp. 443-457.

[12] Gartner Press Release, op. cit., December 13, 2022.

[13] Artificial intelligence assistants are being integrated into traditional software development, for example in Amazon's CodeWhisperer and GitHub's Copilot, and into low-code development platforms such as Microsoft's Power platform. For more information, see: 1), Smith, C. S. *Coding Made AI—Now, How Will AI Unmake Coding?* available at https://spectrum.ieee.org/ai-code-generation-language-models; and 2) Goshal, A. *Why Generative AI Will Turbocharge Low-Code and No-Code Development*, InfoWorld, 21 April 2023, available at https://www.infoworld.com/article/3694173/why-generative-ai-will-turbocharge-low-code-and-no-code-development.html.

the challenges that were faced and the lessons learned, and the anticipated next steps.

Hortilux

Hortilux is a market leader in the development and supply of grow light solutions for international greenhouse horticulture and indoor farming.[14] The company is committed to developing innovative IoT platforms and services to support growers worldwide and maximize results from the use of grow lights.

Hortilux started to develop its HortiSense platform in late 2015 to fulfill the vision of shifting from a product-centric to a service-centric solution provider, as described by managing director, Michel de Wit:

> *"A few years ago, we went through the transition from fixture supplier to grow lighting specialist. … We offer specialized knowledge and support regarding the optimal use of grow lights … to increase the return per [square meter] for our customers through a better use of the available light. We're not just a supplier; we continuously work together with our customers [to achieve] the highest yield, which is expressed in our new slogan: Your Grow Light Knowledge Partner."*[15]

The HortiSense front-end platform was divided into four modules—*grow*, *asset*, *services* and *company*—each representing a different customer-facing service. The grow and asset modules offered IoT-based monitoring and optimization services based on sensor data. The services and company modules allowed customers to plan repairs and maintenance services and display technical and business reports. HortiSense communicated with HortiOffice, a second, back-end platform also developed with Mendix.

Reasons for Adopting Low-Code Development: The main reason for adopting low-code development was summarized by Arno Wartewig, Hortilux's Business Development director:

> *"We are convinced that the IoT is a transformative and unstoppable force. Therefore, we needed an innovation platform that allows us to easily extend our proposition and quickly deliver new functionality. That's why we selected Mendix."*[16]

Combining the Mendix low-code tool with agile development, a Mendix development team was able to develop the IoT platform in a few months. The drag-and-drop graphical user interfaces and declarative programming facilitated the development and allowed these external developers to design applications at a high abstraction level, as Julian Gommers, software developer and product owner of the HortiSense platform, described:

> *"The power of Mendix is that it takes away the difficulty of programming. … On an abstract level, you get a good overview. At university you learn: 'first design then implement.' … It's quite easy to get an understanding of, 'Okay, these are the steps that we will follow in order to get a product.'"*

Gommers also highlighted another advantage of low-code development—the enhanced communication between IT and non-IT people enabled by the common and simple Mendix language:

> *"Sometimes you have a barrier between a very technical person talking in technical terms, and a business person who is not necessarily familiar with those terms. … What Mendix tried to solve here with this low-code platform is basically the translation from design to code. … You design something and you go to your manager and say, 'Okay, well we designed this, and we are going to translate your problem into this solution. Is this okay?'"*

Key Challenges and Lessons Learned: After HortiSense had been implemented, there was a demand to add new functionality, which uncovered the restrictive architectural choices

[14] For more information, see About Hortilux, Hortilux Schréder, available at https://www.hortilux.com/about-hortilux/.

[15] See Harm Ammerlaan new commercial director Hortilux, Floral Daily, November 2, 2020, available at https://www.floraldaily.com/article/9264145/harm-ammerlaan-new-commercial-director-hortilux/.

[16] *Hortilux Drives Business Transformation with IoT-Enabled Greenhouses*, Mendix, available at https://www.mendix.com/customer-stories/hortilux/.

that had been made during its development. These demands revealed the limitations of low-code platforms for developing complex systems. While low-code speeds up development, implementing a platform like HortiSense was a complex task and required "architectural thinking" (i.e., the art of architecting)[17] which, as Fatih Erdogan (a Hortilux developer) described, the initial developers lacked:

> *"The external developers were junior Mendix developers, but not junior software engineers or software developers. They only knew Mendix, and they were asked to build such a big application. If you can't think like a software developer, … then you will have an application that is very complex and very hard to manage."*

Hortilux chose to use Mendix developers because hiring and retaining talent proved difficult. The newness of LCD platforms made it hard for Hortilux to find people with the right skills, whereas software engineers with experience in developing software applications often found LCD unchallenging. As developer Rens Vogels explained:

> *"Mendix was quite new, so you only had so many people who were capable of using it. … That's a problem. You can't get the people. If you [use] Python or C or Java, there are more people [available]."*

Eventually, Hortilux took control of the development efforts. The new internal development team that was responsible for software maintenance and feature extensions quickly discovered architectural limitations and subsequent maintenance issues, as described by Daniel Bos, Hortilux's head of Innovation:

> *"The development in the past years focused on new services and features without keeping an eye on the maintainability and architecture of the apps. The result is that the platforms are difficult to maintain, and the complexity is relatively high."*

Other recurring tasks of Hortilux's software development team involved incorporating regular upgrades of the Mendix tool to its LCD platform. While these upgrades usually provided exciting new functionalities, they often led to the need for rework, as developer Fatih Erdogan explained:

> *"We have functionalities that are not … supported in the newer version. To upgrade to the new version, you first need to rework the current functionality into the new functionality because that's only going to be available in the new version."*

Hortilux's in-house developers unanimously said that a benefit of LCD is that it enabled the development of small applications and proofs of concept. But they also found that developing an IoT platform with multiple services was, as lead software developer, Remy Louwe, described, a much more complex task:

> *"If you want to import something from an Excel sheet and you want to publish it on a website, … that it is doable for a business analyst or people who are not application developers. You can learn it quite easily. But once the application grows in the way that the Hortilux applications have grown, it's not doable anymore for business analysts."*

Next Steps: While Hortilux's software development team set about optimizing the architecture and legacy code of the HortiSense and HortiOffice platforms, senior management began questioning the platforms' future, as described by Daniel Bos, head of Innovation:

> *"More and more time is [needed] to redesign the architecture of both platforms and the associated maintenance of both. … This raises the question [at the senior] management level 'How are we going forward? Should we [re]build it completely … as separated apps or combine them in one app instead?'"*

[17] For more information, see *Architectural Thinking—The Art of Architecting*, IBM, January 2004, available at https://www.opengroup.org/architecture/0404brus/presents/jackson/arch_thinking.pdf.

Hortilux also considered using the Data Hub[18] functionality, a feature announced by Mendix in 2019. Data Hub facilitates the integration of LCD apps with data available from different Mendix applications. While such a centralized solution looked promising, the development team debated whether it was the best approach—worrying that Mendix might eliminate this functionality in the future. Moreover, because Hortilux's future plans included connecting to external platforms, some questioned if low-code was even a viable choice.

Volvo Group

Volvo Group is a Swedish multinational manufacturing company headquartered in Gothenburg, Sweden. It employs over 100,000 people and serves customers in more than 190 markets. The company has 12 different brands and production facilities in 18 countries. As part of its digital transformation, Volvo Group sought to augment work environments with digital technologies and leverage their greater potential by finding innovative ways to solve problems.[19]

One of these innovative problem-solving approaches was LCD. Sebastian Szachnowski, head of Digitalization People Services, described one division where citizen developers, armed with LCD skills, were at the center of Volvo Group's digital transformation efforts:

> *"Within People Services (part of HR Operations), most of the digitalization team consists of people from HR functions. They started in some entry-level HR jobs, but were tech-savvy and wanted to develop digital competences. I have a product owner and developers who started as HR administrators or key users and then started to build competencies to become software developers and build automations. This is a great thing because it is opening a completely new way of developing careers for people, and it is very attractive from the employer's perspective."*

Other citizen developers, distributed across the company and well-versed in business process knowledge, were now accelerating the digitalization of those processes. In doing so, they gained digital competencies and unlocked new career possibilities.[20]

Sebastian Szachnowski described the evolutionary path that low-code and citizen development had taken through the company:

> *"Together with my colleagues from finance back in 2017, we started with the introduction of low-code development in Volvo. First HR and finance deployed the technology, and later it was spread to the whole Volvo Group. Today, we are [still] supporting the citizen development movement, steered centrally by the IT function."*

Reasons for Adopting Low-Code Development: To digitalize processes and services and pursue a company-wide digital transformation, Volvo Group had to find an approach to scale the effort without increasing the demand for support from the IT department. Indeed, the primary motivation for deploying LCD platforms was the limited capacity of the IT department coupled with increasing demands from, and the consequent frustration of, business departments.

One of the main benefits of LCD was that it allowed Volvo Group to quickly upskill and reskill employees, empowering them to develop applications. These employees became citizen developers using LCD to innovate processes that were part of their responsibilities. Some of them even shifted their careers and became professional low-code developers. At Volvo Group, the role of professional low-code developers was project-focused—building applications that fall between the large-scale solutions developed by IT professionals and the smaller solutions built by citizen developers. As Sebastian Szachnowski described:

> *"On the one hand, we upskilled some HR professionals to become IT professionals*

[18] For more information, see: 1) *Mendix Data Hub a "Moon Shot" To Usher in Era Of Unified, Discoverable Enterprise*, Mendix Press Release, April 16, 2019, available at https://www.mendix.com/press/mendix-data-hub-a-moon-shot-to-usher-in-era-of-unified-discoverable-enterprise/; and 2) *Revolutionary Mendix Data Hub Democratizes Data Integration*, Mendix Press Release, September 1, 2020, available at https://www.mendix.com/press/data-hub/.

[19] For information about Volvo Group's digital transformation, see *Digital Transformation Is about People*, Volvo Group, February 12, 2021, available at https://www.volvogroup.com/en/news-and-media/news/2021/feb/digital-transformation-is-about-people.html.

[20] Ibid.

> *that digitalize processes using some more advanced low-code platforms. These are the same people [who] were manually handling these processes in the past. On the other hand, we have other low-code platforms that are used by citizen developers, who are people with no coding or very little coding experience, yet capable of building IT solutions based on low-code. Simply put, professional low-code developers do projects that are too small for the IT department, whereas citizen developers optimize their work or their team's work with applications that would be too small even for the professional low-code developers."*

Key Challenges and Lessons Learned: Volvo Group identified two challenges associated with non-IT employees using low-code to develop applications. First, was the challenge of maintaining the low-code solutions, which required additional resources and competencies, as explained by Sebastian Szachnowski:

> *"If the application is built by professional [low-code] developers, we have resources to maintain those applications. But if it is built by citizen developers in functional roles, then … the manager of that team or department [is responsible for making] sure that they have the capacity and competencies to keep and maintain the app if it becomes part of their processes. … IT is not very willing to take on maintenance of applications they have not built, because they don't know how they were built. It can be a very daunting experience to maintain this kind of application."*

The second challenge of citizen developers using low-code tools was ensuring that applications met privacy and security standards. As Sebastian Szachnowski described, this required offering guidance and education to citizen developers on what they can and cannot do when building new solutions:

> *"Citizen developers must know what they can do and cannot do and, if they have any doubts, they can get support from me or their data privacy representative. In addition, we need more education in this field. When we start processing financial or employee data, we need to be very careful. It doesn't mean that it is impossible [to develop] low-code apps, but we simply need to make sure that we are really compliant."*

In response to the low-code initiative and to institutionalize LCD in the organization, the IT function created a citizen development team. Sebastian Szachnowski described how it supported citizen development:

> *"We have a defined function in IT that is exclusively working with citizen development across Volvo Group. The [team is] supporting the organization regarding citizen development competencies and spreading the idea through the company."*

This reorganization of the IT function was important in supporting and training citizen developers without diverting critical IT resources.

Next Steps: The low-code development initiative to automate business processes is well established across Volvo Group and has the support of the citizen development department. Future initiatives will include the further development of guidelines and education efforts to minimize privacy and security risks.

Volvo Group wants to continue pushing the "citizen" movement, whether through citizen developers, citizen data scientists or other upskilled IT roles, to contribute to the upskilling and reskilling of its employees and equip them with the required digital competencies and capabilities to create innovative digital solutions. As Sebastian Szachnowski described, empowering business employees is seen as the key to moving forward with digital transformation and thriving in the digital age:

> *"I strongly believe that low-code and citizen development, including citizen data scientists, is the way forward regarding the digital transformation of the company. We're digitalizing our company by buying and introducing digital technologies. We roll out digital solutions and we digitalize our business processes. However, the last, and maybe the most important step in this digital transformation, is taking full*

Table 2: Summary of Hortilux's and Volvo Group's LCD Adoption Journeys

	Hortilux	Volvo Group
Focus of LCD	•Development of an IoT platform and services •Digital transformation	•Development of internal processes •Digital transformation
LCD Platform	•Mendix	•Microsoft Power Apps
Main Benefits	•Faster application development •Improved communication between IT and non-IT employees	•Growing the developer base •Talent development through upskilling/reskilling •Balancing IT demand and capacity
Biggest Challenges	•Architectural limitations •Retaining LCD developers •Maintainability	•Security and privacy •Maintainability
Key Lessons Learned	The company must: •Internalize LCD in the organization •Assess task complexity •Build an extendable/reusable LCD platform	The company must: •Establish a department for LCD support •Educate employees on the company's guidelines •Understand LCD use cases and assign users

> *advantage of digital technologies and using them to solve problems in innovative ways. This can happen only through tech-savvy, empowered people. Citizen development and low-code are enabling our people to contribute to this digital transformation and are putting them in the driver's seat."*

Case Comparison of Approaches, Key Challenges and Action Areas

The two cases reveal that implementing LCD is challenging for organizations and requires a dedicated organizational structure and know-how about the technology. Despite making use of different LCD platforms and pursuing diverse goals, both case companies faced comparable challenges and took similar approaches to successfully implementing LCD (see Table 2 for a summary comparison).

Both companies identified similar reasons for and benefits of adopting LCD, the main one being the need to balance demand and capacity in application development. Hortilux had initially adopted LCD to deal with the lack of developers in the organization and because of the development speed and flexibility it provided. Volvo Group had also adopted LCD to balance the increasing demand on the IT department with capacity. Furthermore, Hortilux also viewed LCD as a means to facilitate communication between technical and nontechnical employees. Volvo Group went one step further: not only did LCD improve communication but it also enabled the upskilling and reskilling of nontechnical employees, allowing them to become citizen developers who could build their own applications.

The challenges identified in both cases point to the importance of providing a low-code platform architecture that facilitates the maintenance of the solutions. They both highlight the challenges of understanding the technical constraints of the architecture, retaining talent, educating employees on issues of security and privacy, and providing institutionalized support for LCD.

Recommendations for Adopting Low-Code Development

The interviews we conducted with the two case organizations as well as executives at the seven additional organizations helped us identify questions in five action areas that CIOs, CDOs and

Table 3: Five Key Questions to Address When Considering Adoption of LCD

Strategy: How should an organization identify LCD opportunities?
Players: Who should be involved in LCD efforts? What skills and knowledge do they need to effectively participate in LCD?
Organization: What organizational factors enable LCD activities?
Infrastructure: What infrastructure elements are important for LCD?
Solutions: What technical requirements is low-code best suited to meet?

business leaders considering LCD should address (See Table 3).

Below, we provide five recommendations that will help CIOs, CDOs and business leaders successfully adopt LCD. Each recommendation relates to one of the five key questions listed in Table 3. All of our interviewees unanimously rated these recommendations in a follow-up confirmatory survey as having moderate to high relevance to low-code-based digital transformation.

1. Set a Clear Strategy for Selecting LCD Use Cases

This recommendation helps to answer the strategy question: *How should an organization identify LCD opportunities?*

Business leaders need to be aware of the opportunities and limitations of LCD. As with any other technology, before implementing LCD, leaders need to ensure that their organization clearly understands the potential use cases and is ready to implement them. An executive at industrial manufacturer InCo (the pseudonym of one of the seven other organizations included in our research) highlighted this point as an important issue in any low-code adoption journey:

> *"Finding the right use case is still the biggest challenge, … understanding the business logic and the business model. I would say that's one step before you start implementing [a low-code platform]."*

When programming with low-code, the coding is done on a higher, more abstract level. While this makes low-code attractive for non-IT specialists, professional developers might view this as a limitation for working on complex tasks because tracking the performance of the code on a higher level becomes harder. Using a platform that supports the transition from low-code components to traditional code (e.g., C++, C, Python or Java) and vice versa could address this issue and allow professional developers to develop advanced solutions, boosting low-code performance.

An executive at TechCo (another pseudonym—see Appendix) mentioned the importance of periodically revising the LCD strategy as the development platforms and opportunities continuously evolve:

> *"We are constantly reviewing the latest development tools. [For example] we are exploring the use of [low-code analytics platform] Knime. In production systems, we are mainly using R and Python. Knime helps internal customers (such as development engineers, quality management and marketing) understand what we do and enables them to develop their own use cases based on the data and interfaces we provide. We are constantly working on our [LCD] vision."*

As citizen developers are playing a more prominent role in the development of applications, business leaders need to define clear strategies for selecting the use cases that are most suitable for using low-code and ensure that their employees have clear guidelines.

Because of the characteristics of low-code development—its plug-and-play character, quick assembly and high abstraction level—it's best to start with simple, low-complexity tasks, such as process automation and small proofs of concept. This is particularly important when people without IT expertise are involved and when LCD is first implemented in the organization.

2. Identify and Assign LCD Users Who Can Benefit Most and Provide Training to Upskill/Reskill Tech-Savvy Employees

This recommendation addresses the players' questions: *Who should be involved in low-code development efforts? What skills and knowledge do they need to effectively participate in low-code development?*

CIOs and CDOs need to assess which business functions and users can benefit most from LCD and ensure that employees in those functions are empowered and supported. This assessment should distinguish between tech-savvy citizen developers and professional low-code developers.

For tech-savvy citizen developers, CIOs and CDOs should prioritize business departments with a high demand for automating processes and with low-complexity applications—for example, many early adopters such as Volvo Group first introduced low-code development in the finance and HR departments. Business leaders in these selected departments should identify tech-savvy employees interested in upskilling to become low-code citizen developers. Once citizen developers are assigned, organizations need to provide the right training and support to upskill these employees and ensure they are aware of the opportunities of LCD, its requirements, and the privacy and security policies that should inform any development. Ultimately, employees should have clear accountability for the LCD applications they build, irrespective of their technical expertise.

CIOs and CDOs can also benefit from creating a team of professionally certified low-code developers who can develop smaller applications and proofs of concept and thus mitigate increasing demand on the IT department and speed up the development of applications.

In some cases, low-code development involves a hybrid arrangement where business and IT experts work together. Because of this, in addition to assigning LCD users in the organization and providing training, business leaders should ensure adequate collaboration and understanding across the diverse LCD players. They might also involve customers early and throughout the LCD process, as highlighted by the innovation manager and low-code owner at EngCo (another pseudonym):

> *"The relationship with the customer needs to change as well. With low-code, you can help the customer get a feeling for what they asked for and receive feedback quickly."*

3. Establish a Low-Code Team or Department with Professional Low-Code Developers Who Provide Organization-Wide Support and Training

This recommendation addresses the organizational question: *What organizational factors enable low-code development activities?*

CIOs and CDOs should create a team or department focused on low-code to institutionalize and legitimize the LCD approach and provide resources for adopting it across the organization. This organizational unit should consist of certified professional low-code developers who have the skills and expertise to benefit from the full potential that LCD offers in terms of development speed and flexibility. By developing simple applications and proofs of concept, professional low-code developers can free up resources in the IT department and increase the organization's developer base. An executive at AuCo (a pseudonym) reminded us that it is possible to upskill nontechnical employees into professional low-code developers, thus helping to tackle the application development bottleneck:

> *"Usually, when you look at IT organizations, there's not the right skill or not sufficient capacity to fulfill all the business demands. … Because [low-code] is easy to use, … anybody can learn it quickly."*

Moreover, as LCD experts, professional low-code developers can provide support for upskilling and reskilling business employees to create a cohort of citizen developers. Tasks involved in supporting citizen developers include offering guidance to identify LCD opportunities to progress digital transformation in business departments, providing guidelines to meet privacy and security policies in LCD applications, and supporting the maintenance of applications developed in business departments.

4. Ensure the LCD Architecture Includes Extendable and Reusable Components and Is Up to Date

This recommendation addresses the infrastructure question: *What infrastructure elements are important for low-code development?*

CIOs and CDOs should ensure that business functions have relevant knowledge about the architecture of digital solutions and that business leaders enforce and promote architectural thinking in the applications developed in their departments. Though the high-level abstraction of low-code-based applications provides a logical approach to developing applications, employees using LCD tools might underestimate the relevance of having a clear architecture that is easily extendable and reusable. Such an architecture will prevent overly complex, siloed and inefficient applications from being built. Architectural thinking is therefore even more important for business functions that have citizen developers or professional low-code developers who do not have a software development background.

At the same time, business leaders need to ensure that the LCD platform architecture is flexible and modular so that it can easily be extended and reused in the development of future solutions. It is also important to keep the low-code components (code snippets) of the platform well-maintained and to ensure that the components are updated in line with technological advancements. As described by a solution manager at TechCo, the rapid pace of technological advancements means that organizations must revise their low-code components regularly to incorporate new developments (e.g., a new communication protocol), thus ensuring that the components are not outdated:

> *"An important aspect is the maintenance. How do you keep the platform up to date, and how do you support new hardware that is coming [to market] even faster than software?"*

Finally, to enable LCD efforts and avoid siloed developments, business leaders should ensure that data is accessible and aggregated across different systems, portfolios and domains. Along with reusable low-code components, the availability of aggregated data facilitates the development of digital solutions that streamline processes and create new forms of value for employees and customers.

5. Evaluate the Technical Requirements and Constraints of Solutions and Identify which Low-Code Platform Is the Best Fit

This recommendation addresses the solution question: *What technical requirements is low-code best suited to meet?*

When adopting LCD, CIOs, CDOs and business leaders need to assess the technical requirements of the solutions and select which LCD platform is the best fit. First, it is important to assess the use case at hand and whether LCD can meet the technical requirements and constraints of the desired solutions. For example, a software architect at TechCo highlighted a potential problem of using low-code to develop IoT applications for products with embedded IoT sensors. Because such products might have performance constraints (e.g., running on a low-power battery for a long time), organizations might want to avoid low-code or only use it for developing a proof of concept in an emulated environment:

> *"We develop complex IoT devices that need to run on a battery for 10 years. How do I write such a code with low-code blocks?"*

Second, once the applicability/suitability of LCD has been assessed, CIOs and CDOs should evaluate which LCD platform best fits their needs. The heterogeneous IT landscape and the varying needs across organizations might persuade them to use multiple platforms. For example, a firm that has a license for Microsoft Power Apps or a Salesforce MuleSoft might want to use the associated LCD platform to enable citizen developers to implement simple process automations. However, such a firm might also adopt more technical LCD platforms such as Mendix or OutSystems to enable professional low-code developers to build more advanced applications.

The wide range of available LCD platforms makes it challenging for companies to choose the right platform. They should consider the following two questions when deciding which platform best suits their needs: *1) Do our requirements need a general purpose or a domain-specific platform? 2) Do our LCD efforts primarily involve professional or citizen developers?* There are many general-purpose platforms such as Mendix and OutSystems, but there are also some domain-specific platforms, such as Alteryx, which offer end-to-end automation of data analyses and machine learning. Moreover, platforms such as Mendix and OutSystems are more focused on professional development and support more advanced functions, such as software testing and scrum project management. These platforms are better suited for professional users because citizen developers would face a longer learning curve before they can make optimal use of them. Platforms such as Microsoft Power Apps or Google App Maker are better suited for citizen developers because they are more intuitive to use and require fewer competencies and skills.

Finally, companies should consider the state of development of potential platforms, which are continuously changing. As low-code platforms evolve and "artificial intelligence and machine learning … hide code complexity"[21] while delivering on code functionality, the distinction between simple LCD environments for citizen developers and advanced environments for professional developers will become blurred.[22]

The Five Recommendations are Interdependent

The five recommendations provided above are highly interdependent, and business executives and leaders exploring the adoption of LCD should use them in combination. For example, the selection of use cases for LCD (Recommendation 1) should also include the evaluation of technical requirements and constraints (Recommendation 5). The use cases the organization first contemplates may overly influence the choice of LCD platform, which will in turn create technical constraints. Furthermore, the initial users identified for LCD upskilling (Recommendation 2), will influence the makeup of the professional low-code development department (Recommendation 3). Moreover, this department will identify the use cases that the company opts to explore (Recommendation 1), the kinds of training offered to support citizen developers, and the set of workers who show an inclination to upskill and become citizen developers (Recommendation 2). Finally, the selection of an LCD platform (Recommendation 5) will influence the development of a clearly defined architecture that limits or enables the reuse and extension of low-code components. Thus, as organizations gain competence in LCD, they will need to action all five recommendations concurrently. This will enable them to continuously match organizational low-code capabilities with opportunities for exploiting LCD tools and platforms.

Concluding Comments

Using low-code development can fuel digital transformation by helping organizations tackle the application development bottleneck created by insufficient capacity in centralized IT departments. Low-code can help organizations automate and optimize processes and foster a culture of innovation that involves diverse groups of experts. It enables nontechnical players to be involved in application development and increases customer orientation, thus reducing the cost of managing increasing digitalization, creating a culture of digital change, and offering opportunities for transformation across the organization.

LCD facilitates these changes in different ways. First, the visual and simple syntax of LCD tools and their modularity allow users with more diverse backgrounds to participate actively in the application development process. With LCD, the role of non-IT-savvy employees (i.e., citizen developers without a software development background) in the application development process changes from contributors (i.e., information resources) to a more active role as process innovators, application designers and codevelopers. Second, LCD accelerates change by enabling the plug-and-play assembly of low-code development components that speed up the development of prototypes.

For organizations to extract the most value from LCD, they need to understand low-code platforms, the requirements of the solutions and the required changes to organizational processes. It is important they follow best practices to select

[21] Carroll, N. and Maher, M., op. cit., June 2023.

[22] Ibid.

and deploy the right low-code platform and that they learn from early adopters pursuing different use cases. In this article, we have presented the cases of two early LCD adopters—Hortilux and Volvo Group—and our findings from interviews with executives and technologists from these two companies and seven other organizations. From this research, we have extracted and confirmed five interdependent recommendations for adopting LCD. Following these recommendations will enable organizations to successfully adopt the low-code approach and take advantage of the multiple benefits that LCD brings for fueling digital transformation.

Appendix: Research Methodology

Given the early stage of most firms' use of LCD for fueling digital transformation, our research study was based on in-depth cases of two successful early adopters of LCD as well as semi-structured interviews with executives in seven other organizations using LCD. To identify actionable recommendations for future adopters, we took an action principles approach.[23] Action principles research studies innovators and early adopters to identify relevant action principles that can guide the implementation journey of

Overview of the Interviewees

Firm	Industry	Number of Employees	Interviewees and Roles
Hortilux Schréder b.v.	Horticulture lighting	Over 50	-Daniel Bos, Head of Innovation -Wim Alsemgeest, Technical Engineer Automation -Rens Vogels, Embedded Hardware and Software Engineer -Julian Gommers, Software Developer and Product Owner HortiSense -Fatih Erdogan, Software Developer -Remy Louwe, Lead Software Developer (External)
Volvo Group	Automotive	Over 100,000	-Sebastian Szachnowski, Head of Digitalization People Services
Pseudonyms: AuCo	Automotive	Over 200,000	-Team Lead of Digital Solutions
EleCo	Consumer electronics	Over 1,000	-Director IoT Products
EngCo	Civil engineering	Over 8,000	-Innovation Manager and Low-Code Owner
GarCo	Consumer durables	Over 10,000	-Director of Digital Products
InCo	Industrial manufacturing	Over 350,000	-Head Data Science
LiCo	Lighting	Over 35,000	-Director of Innovation
TechCo	Engineering and technology	Over 400,000	-VP IoT Products -Head of Engineering IoT Products -Solution Manager

[23] For more information on the action principles approach see, Lacity, M., Willcocks, L. and Gozman, D. "Influencing Information Systems Practice: The Action Principles Approach Applied to Robotic Process and Cognitive Automation," *Journal of Information Technology* (36:3), September 2021, pp. 216-240.

future adopters and borrows from the critical success factors (CSF) method.

The fieldwork for this study was conducted during the first half of 2021 (Hortilux) and 2023 (Volvo Group), when we collected data about both companies' LCD implementation journeys. We also collected data from seven additional organizations via exploratory semi-structured interviews. In total, we interviewed 16 executives and technologists responsible for implementing low-code initiatives in these nine companies. Interviewees were asked about their LCD adoption journeys and provided with information about LCD benefits, drivers, use cases, challenges and practices. The table above provides an overview of the interviewees and their firms. Some of the quotes included in the article were taken from the interviews and have been slightly modified for readability.

We complemented the fieldwork with an online survey questionnaire to further explore and confirm the relevance of the derived recommendations. While this is not part of the action principles approach, it is common practice in CSF research and can contribute to a better understanding of the relevance of the action principles/guidelines.[24] The questionnaire was completed by eight interviewees who rated the perceived importance (i.e., criticality) of each recommendation on a five-point Likert scale: 1 = *not at all important*; 2 = *slightly important*; 3 = *moderately important*; 4 = *very important*; and 5 = *extremely important*.

About the Authors

Ainara Novales

Ainara Novales (novales@rsm.nl) is a researcher at Rotterdam School of Management, Erasmus University, the Netherlands, and a senior management consultant at adesso SE, an independent IT service provider in Dortmund, Germany. She investigates the impact of digitalization on products and services as well as on sustainability. Her research interests include the Internet of Things, digital innovation, low-code development, platform ecosystems and digital sustainability. She has published her work in multiple international conference proceedings, and in 2021 received the Best Paper Award at the CIO Forum of the International Conference on Information Systems (ICIS).

Rubén Mancha

Rubén Mancha (rmancha@babson.edu) is an associate professor of information systems in the Operations and Information Management Division at Babson College, Boston (U.S.). His research focuses on digital transformation and digital sustainability, and his work has been published in journals including *Decision Support Systems*, *Journal of Business Strategy*, *Information Technology & People* and *Journal of Environmental Psychology*. Rubén teaches information technology and digital innovation courses at the F. W. Olin Graduate School of Business and advises digital startups and companies undergoing digital transformation. In 2017, he was recognized as one of the Best 40-Under-40 Business School Professors by Poets & Quants.

[24] See for instance: 1) Bhatia, M. S. and Kumar, S. "Critical Success Factors of Industry 4.0 in Automotive Manufacturing Industry," *IEEE Transactions on Engineering Management* (69:5), September 2020, pp. 2439-2453; and 2) Remus, U. and Wiener, M. "A Multi-Method, Holistic Strategy for Researching Critical Success Factors in IT projects," *Information Systems Journal* (20:1), January 2010, pp. 25-52.