CIO ASSOCIATION OF CANADA

Low-Code Development for CIOs

Essential Recommendations for IT Leaders

DAZZM

Introduction

In 2011, the growing value enterprises were experiencing through advanced application development led Netscape co-founder and venture capitalist Marc Andreessen to pronounce that software was "eating the world." Since then, IT leaders have had to be increasingly adept at ensuring the appetite for greater digitization doesn't eat up all their resources in the process.

The increased demand for application development comes from many directions. Where CIOs might have once focused their developers on operational applications that support the back office, the quest to enhance customer experiences has prompted the call for more ways to engage digitally. This not only includes e-commerce processes on an organization's own website but applications that connect to social media channels, third-party marketplaces and more.

At the same time, increased digitization in our personal lives raises the bar for the tools employees expect to use at work. IT leaders may find themselves fielding ideas and requests for applications that could transform sales, marketing, customer service, HR and much more.

Low-code development platforms represent the only viable way to address these challenges. Among other benefits, these platforms allow those on the front lines of business to play a more active role in creating the software that runs the business. In fact, according to market research firm Gartner Inc., the proportion of low-code application users who are not in IT departments will grow from 60% in 2021 to 80% in 2026.*

That said, low-code (and no-code) development raises several questions that IT leaders must consider from a change management perspective. This white paper aims to help guide them by providing recommendations on some of the most critical areas. Read this to learn:

- Business catalysts that will drive a greater need for low-code development
- How to prioritize use cases from a risk-based perspective
- The promise of citizen development and the guardrails needed to ensure success
- Best practices for driving alignment between business and IT through low-code development
- What low-code means for the future of developers and their roles within the business

The insights you'll read here are complemented by anonymized comments from real IT leaders from the CIO Association of Canada (CIOCAN) who participated in a virtual roundtable sponsored by DAZZM about their low-code development journey to date.

Low-code and no-code platforms represent a way forward for CIOs who are trying to bridge the gap between ideation and execution across the enterprise. There is no better time to get started than right now.

Forecast Analysis: Low-Code Development Technologies, Worldwide. Gartner, 2022

1. Modernizing applications at the speed of business

The challenge

CIOs have always been responsible for keeping applications and platforms up to date in order to drive optimal performance and avoid security issues or other vulnerabilities. The interest in digital transformation among senior executives is now putting greater pressure on IT leaders to lead a more comprehensive and ongoing modernization effort, which can make planning and prioritizing more difficult.

What CIOs say

"There's an expectation of IT to move fast, yet business still moves slow. IT governance becomes really important to ensure that you're managing the resources in the most effective way, and yet moving the organization forward at the same time."

Recommendations

Staying on top of modernization – and even getting ahead of it – means looking beyond the traditional catalysts for change.

For example, if you're focused on data governance, you're probably already aware of the risks associated with current solutions that might be no longer in compliance with particular laws or regulations. It's time to step back and take a more holistic look at your application portfolio to prepare for its future evolution.

This includes:

- **Identify the makeshift solutions that could make more work for IT:** Employees might have tried to jerry-rig their own digital solutions using tools such as Microsoft Access or Excel. These only tend to work based on the expertise of those who created them, however, and there's no roadmap for maintaining and improving them. These are likely where low-code development could provide a much more viable alternative.

- **Take every possible cloud migration into account:** Some applications may be running in isolation from others and lack the benefits that come from integration. Moving those applications to the cloud may not be possible without first rewriting them, though. Be sure to explore how low-code development could bridge this gap.

- **Confront projects that have been placed on the back burner:** Some applications may have needed to evolve for some time but rewriting them has seemed daunting. Talk to a trusted partner about using low-code development to accelerate this process.

"There's an expectation of IT to move fast, yet business still moves slow. IT governance becomes really important to ensure that you're managing the resources in the most effective way, and yet moving the organization forward at the same time."

2. Building business cases and achieving buy-in

The challenge

CIOs may be familiar with low-code development and believe it can solve their challenges, but they need to get support from the top. However, senior executives may be reluctant to shift away from the status quo or are more focused on cost containment and other strategic issues. IT leaders need to be able to articulate a vision for the future of application development based on low-code approaches that will resonate with the C-suite.

What CIOs say

"I think what really differentiates organizations who can meet the business needs is having a portfolio view of what their investments are going to be and having the business involved to understand what needs to be done."

Recommendations

The overhead associated with traditional development can sometimes kill a project before it even begins. Senior executives may have had bad experiences with existing applications where tweaking even a single workflow based on a particular use case could have required 20 developers.

Your role as a CIO is to start an education process that helps the C-suite understand how low-code development uses declarative and graphical techniques that minimize traditional, labour-intensive coding processes. Make sure to also touch on:

- **The concept of composable architecture:** Gartner and others have predicted that many businesses will embrace application design based on reusable components to build applications more quickly and easily. Help senior leaders see what this means in terms of the resources required for projects.

- **The change in application lifecycle:** Many businesses have at least some software that's been running for decades. Explain how low-code approaches allow applications to be upgraded or replaced in much shorter timeframes, creating greater business agility.

- **The move away from monolithic applications:** The technical debt businesses carry is usually based upon large legacy applications that are difficult to change. Your conversation about low-code development needs to shift the thinking around digital transformation. Instead of one huge project, low-code tools allow you to work on dozens of smaller projects that can all connect and disconnect from each other as needed to provide greater value.

"I think what really differentiates organizations who can meet the business needs is having a portfolio view of what their investments are going to be and having the business involved to understand what needs to be done."

3. Secrets of successful citizen development

The Challenge

While low-code platforms and tools democratize the ability to create valuable business software, there's a balance to be struck between offering autonomy and enforcing governance. CIOs need to ensure nothing compromises the integrity of the architecture, data protection and overall integration across a portfolio of applications. IT leaders also need to contend with limitations in development expertise among business users and the long-term need to scale.

What CIOs say

"Citizen development scares the hell out of me from a security data and privacy standpoint … people move around the organization, and you can suddenly have a problem mushroom, where it's like, 'Oh, my God, this application is not working – get somebody in IT.' Then IT looks at it and says, 'What the heck is that stuff?'"

Recommendations

Encouraging citizen development does not have to come at the expense of increased cyber threats, siloed applications and maintenance issues. It's just important to remember that even with low-code platforms and tools, the need for highly skilled IT professionals doesn't necessarily go away. This is more like an evolution of the partnership between business and IT. Therefore:

- **Ensure IT owns the tools first.** Make sure your existing developers and any other relevant stakeholders within the IT department become proficient with low-code platforms and have some experience developing applications with them. This will help when the time comes to provide guidance and coaching for line of business professionals.

- **Involve citizen developers without necessarily expecting them to do everything.** Business users are the most familiar with the complexities and nuances of their particular workflows and tasks. They might therefore be best to take part in application development using low-code tools to create pilot projects or proof of concepts, while more experienced developers could build them out further. In other cases, developers might build an application but citizen developers could add modules as new business needs emerge.

- **Use increased visibility to showcase the path to scalability.** When a business user creates a DIY solution in an Excel spreadsheet, there's no way to track and monitor it at an enterprise level. The best low-code development platforms provide dashboards that offer this kind of visibility, which allows IT and the business to manage software assets and ensure they fit within the existing enterprise architecture. Decentralizing application creation doesn't mean you're abandoning the need to manage your application portfolio. In fact, the opposite is true: with low-code development it finally becomes possible to effectively oversee hundreds of applications and understand details such as their lifecycle.

- **Look for projects that are relatively self-contained.** Citizen developers should champion the creation of software that is used by their team or department, but they should not lead one that is highly technical in nature or could affect any interdependencies with other platforms or applications connected across the entire enterprise.

- **Establish a clear process for the new approach.** Citizen developers may be reluctant to get involved with application development at first because they are accustomed to projects where IT departments needed six months or longer to go into production, and only after they've received lengthy technical specifications. Help business users realize that low-code development should shorten some projects to a matter of weeks. Also talk about ongoing development, such as creating new iterations on a regular basis such as twice a week or even twice a month.

- **Treat citizen development projects as ongoing works in progress.** Business users may be excited about the speed of execution they gain through low-code tools, but they should also understand that the end results of their work will morph and adapt in response to business needs. You don't have to completely map out every application process on a whiteboard. Teach them about the thinking behind agile development and how the applications they help create will grow in sophistication and value over time.

- **Lean on trusted partners and industry best practices.** Your low-code development technology provider should be able to share what's worked and what hasn't from a citizen development perspective. When you invest in their tools you're also gaining their expertise, so make good use of it. There has also been some great work done in this area by organizations such as the Project Management Institute (PMI), which offers a set of vendor-agnostic educational resources that could benefit those working in IT and line-of-business functions alike.

- **Stick to the fundamentals.** It doesn't matter whether you're involving citizen developers or those with a deep programming background. Any project you undertake should still be subject to code review, system testing, integration testing and user acceptance testing before anything goes live.

4. Taking a risk-based approach to low-code development

The Challenge:

Applications that fail or don't perform as expected aren't merely annoying. They can lead to businesses losing customers. It can open the door to rogue actors who spread malware and other cyberattacks. Storing personal information unsanctioned or makeshift solutions could lead to files being compromised, headline-making data breaches and financial repercussions, among other consequences.

Poorly executed application development projects can also lead to increased costs, compliance concerns and employee turnover. No organization can afford to overlook the potential consequences.

What CIOs say

"You basically map out, what's the likelihood that something's going to happen? As a CIO, you have to get your data to a point where you have the proper disaster recovery and backups in place."

Recommendations

Digital transformation should come hand-in-hand with a closer alignment between IT and the business. Only by having a constructive dialogue can both CIOs and their counterparts in other functions balance the benefits of digitizing with the potential downsides. Do this by:

- **Map out the scope of risk:** The truth is, relying on makeshift solutions developed in Access or Excel are already creating risks, as we've discussed earlier in this whitepaper. They might not be obvious at first but can lead to expensive headaches later on. New projects may take businesses into uncharted territories, but the risks involved may be lesser than sticking with traditional monolithic application development approaches.

- **Achieve consensus around acceptable risk:** Some projects are so important to the business – due to revenue implications or customer expectations, for example – that they may cause a possible but unlikely negative outcome. Both IT and the business should agree on this area and quantify it in any way they can. This is also where you can do more from a testing and/or governance perspective to mitigate risks to some extent.

- **Have a plan for quick response and remediation:** The risks in traditional application development could be arguably higher in some cases because changes were not easy to make. Low-code platforms and tools change that, and it's just another reason IT departments should own the tools and understand how to make use of them to solve problems in urgent situations.

"You basically map out, what's the likelihood that something's going to happen? As a CIO, you have to get your data to a point where you have the proper disaster recovery and backups in place."

5. Leading dev teams amid the low-code transition

The Challenge:

Professionally trained programmers may be suspicious of low-code technologies, and even threatened at the prospect of citizen developers who take application projects into their own hands. CIOs have to pursue decentralization without having an adverse impact on developer retention and make sure their talents can be focused on higher-value or more challenging work.

What CIOs say

"(Low-code) is more about empowering developers so that they hopefully become more focused on the functionality that we need to build, not the code that's required to implement that functionality."

Recommendations

Many developer teams are made up of people whose greatest passion is to find the right solution to business problems, and they recognize that code is just a means to do so. IT leaders have to make them feel included in the shift to low-code development and as empowered as those in functional areas. Here's how:

- **Make the 'screwdriver vs. a drill' distinction:** In the construction sector, contractors and their teams can use a rudimentary tool like a screwdriver to put screws in a wall. With the advent of power drills, however, a lot of those chores were dramatically sped up. Low-code development offers a similar advantage, even to developers who see coding as a craft.

- **Demarcate the areas where low-code is likely to apply:** Projects that require deep knowledge of application programming interfaces (APIs) and other technical components will continue to be within the purview of professional developers. Those more focused on gathering and presenting business data are better suited to citizen developers. Help the team see where they will continue to use their traditional skills.

- **Re-engage functional analysts:** Some employees start out as developers but tire of getting mired in reviewing technical specifications and translating them into lines of code. While you may have to spend some time helping longtime developers dispel misconceptions about low-code tools, these same technologies could help those in other roles get excited about creating software again. And that enthusiasm could drive greater cooperation and morale among developers and other teams across the organization.

"(Low-code) is more about empowering developers so that they hopefully become more focused on the functionality that we need to build, not the code that's required to implement that functionality."

Next steps: Embracing low-code development with confidence

The five areas we've covered in this white paper can help you avoid barriers to using low-code platforms and tools while also ensuring all stakeholders are positioned to use these technologies to their maximum potential.

If you've read this far and are ready to move forward, there are three other questions you should ask yourself and your leadership team:

1. "Should we establish a low-code center of excellence?"

Traditional developers often have to say "No" to new projects because they're already incredibly busy with highly technical projects that make good use of their skills and expertise.

Some organizations set up a low-code CoE (Center of Excellence) because they recognize the value in having a dedicated group that will work on smaller, less mission-critical applications that require a different approach and mindset. These can be cross-functional teams that come together based on specific or time-sensitive business needs and use low-code tools and engage citizen developers as required.

2. "How far should we be on our cloud journey?"

Traditional software development has been run on costly, often difficult to manage servers and other on-premise infrastructure, along with licensing expenses. The move to low-code development offers a chance for organizations to also take advantage of a cloud-based approach that makes more sense for smaller projects with minimal technical constraints.

Although many organizations have at least begun moving infrastructure and data to the cloud, embracing low-code development could add another layer to their business case. It might also call for a partnership with a trusted third party that can assist with cloud migrations so that organizations can stop running on local servers even sooner.

In doing so, organizations can benefit from offloading application runtime and management to a trusted vendor. Low-code platforms not only speed up and simplify development but eliminate the need for traditional monitoring and maintenance chores which can take up a lot of the IT department's time.

3. "Are we choosing the right platform?"

Some CIOs have had previous experience with low-code tools from years past, but they may not have explored the best the market offers today.

While organizations will evaluate providers from a variety of angles, a best-in-class low-code platform should be based on an open architecture with business composability in mind, as well as drag-and-drop capabilities and organized business logic management.

Conclusion

Digital transformation will continue to be the means by which organizations elevate their workforce, deliver a better customer experience and achieve their most critical business outcomes.

CIOs will continue to help people stay connected and collaborate, for example, as they adopt hybrid work models. The rise of virtual and augmented reality will open up new avenues to engage with customers. AI will force everyone to reimagine everyday processes.

All of these trends – as well as those we can't anticipate – will be driven by applications that need to be developed, tested and put into production at an unprecedented pace. Yet speed can't come at the expense of quality, security or effective governance.

CIOs are excited about the potential of low-code and no-code platforms for the same reasons they have been embraced by their early adopters around the world.

When coupled with the right practices and policies, low-code tools are helping organizations act with the agility their customers expect.

They can help IT teams rationalize applications and make better use of resources.

Low-code platforms also solve the build vs. buy conundrum CIOs often face, and enable continuous delivery and integration.

Most importantly, low-code technology could help unleash innovation everywhere, and allow IT leaders to bring more great ideas to fruition. That's what helps CIOs feel most fulfilled in their role, and demonstrates the unique value they contribute to the business.