



Intellyx™



White Paper

Secrets to Low-Code Success in the Enterprise

Jason Bloomberg

President, Intellyx

June 2019



Getting started with a low-code platform by tackling a single application or proof of concept is straightforward. Moving beyond that first app toward enterprise low-code deployment, however, is more difficult.

Among the challenges: enterprises must avoid shadow IT issues while integrating new apps with existing applications, both on-premises and in the cloud.

As such organizations proceed with their legacy modernization efforts, low-code platforms like Kony Quantum can lower the risks inherent in such efforts, while accelerating their progress.

Low-code platforms can also be instrumental in reducing application backlog, while becoming an integral part of both the technology and organizational change that constitute enterprise DevOps.



Beyond the First Low-Code Project

Of all the disruptive trends impacting enterprise technology today, among the most impactful to hit the application development world are low-code platforms. Low-code platforms accelerate and simplify the work of professional developers, empowering them to focus on building better software more quickly – software that can be as complex or sophisticated as the enterprise requires.

In spite of this potentially unlimited sophistication, however, many low-code application creators focus their efforts on what we might call discrete applications: applications that are for the most part self-contained, running primarily or entirely within the context of the low-code environment.

Such applications can meet many enterprise needs to be sure, but challenges of security, compliance, and integration remain. To address these issues, low-code platform vendors are rising to the occasion, implementing enterprise-class solutions to each of these areas of concern.

Many low-code application creators focus their efforts on what we might call discrete applications: applications that are for the most part self-contained, running primarily or entirely within the context of the low-code environment.



As a starting point, however, many organizations begin their low-code journey with a single application. Sometimes the first application is a proof of concept, intended to try out the platform and get the development team up to speed. In other cases, a line-of-business (LoB) executive is the low-code champion, and as a result, the first app addresses a departmental need.

Given the speed, cost, and quality benefits of low-code application development, however, it doesn't take long before organizations move beyond the single-app phase



and begin to roll out multiple applications. Early quick wins lead to broader acceptance, and ongoing successes typically lead to a clamor of demand, as individuals across the enterprise hear of the newly deployed apps.

Soon, such applications form an ecosystem. Instead of a disparate set of disconnected apps, low-code platforms encourage consistency, interaction, and integration of applications, even across lines of business (LoBs). The result is improved support for strategic, customer-driven digital efforts, as users both inside and outside the organization increasingly expect and demand coherent, well thought-out sets of applications from the companies they do business with.

Navigating Past Shadow IT Pitfalls

As organizations reach this multiple-application phase, the change management capabilities of the low-code platform become increasingly important. Organizations must establish a solution delivery governance framework that allows – and encourages – rapid development of applications across LoBs.

Without such governance, such frameworks are difficult to implement in practice. The end result is a measure of chaos, as different LoBs find that centralized IT isn't able to keep up with their needs. In many cases, this chaos takes the form of shadow IT: when LoBs bypass the IT department in order to roll out applications quickly.

The causes of shadow IT are well-known: the IT organization is too slow and process-bound to respond quickly enough to the needs of LOBs, so business users go around IT and buy – or build – their own apps and other technology.

The problems of shadow IT, unfortunately, are also quite familiar: without proper oversight and coordination, security vulnerabilities and compliance violations can proliferate, with no one at the helm to address such issues.

Furthermore, these citizen-built apps can be redundant, obsolete, or otherwise low quality. We like to call this the 'Lotus Notes' problem, many organizations implemented ungoverned apps using the popular Notes platform during the 1990s and 2000s, and many such legacy apps remain in place.

For modern no-code platforms (intended for LoB 'citizen' developers), this shadow IT pitfall is still a risk. Certainly, some of the more mature low-code platforms (aimed at professional developers) tackle the shadow IT issue head on, ensuring that apps on



these platforms are adequately secure, compliant, and address ongoing business needs in a lightweight, business-focused manner.

Integrating Low-Code with Existing Apps

While early low-code apps tend to be standalone, it doesn't take long until LoBs want to integrate their apps with other applications in the enterprise and in the cloud. Without the proper low-code integration capabilities built in, such integration presents a difficult set of challenges, especially for more complicated integration requirements.

It doesn't take long until LoBs want to integrate their apps with other applications in the enterprise and in the cloud. Without the proper low-code integration capabilities built in, such integration presents a difficult set of challenges, especially for more complicated integration requirements.



The good news: today's low-code platforms all generally support APIs, both creating them as part of the low-code application building process, as well as consuming them from third-party endpoints, including on-premises applications, databases, and SaaS apps like Salesforce and ServiceNow.

As low-code deployments become more complex, organizations may focus on building automated, multi-step data orchestration. They soon find that there is more to the integration challenge than simply connecting APIs.

In such situations, the application back-end itself requires some kind of hands-on development, either to expose it with APIs, or in the more general case, to modernize it to conform to broader architectural requirements like the horizontal scalability and elasticity we expect from the cloud.

The low-code platform's focus on discrete applications then falls short of the needs of the organization. It may be straightforward to build the front-end of a multi-tier



application using low-code, but shifting the back-end, say, to a containerized microservices environment would still fall to hand-coding.

This low-code front-end/'high-code' back end situation is evolving, however, as more low-code vendors tackle back-end integration from a low-code perspective.

Delivering back-end capabilities as modular microservices is typically an essential part of this evolution, especially when vendors expose those microservices as APIs and include them in their platforms' declarative, drag-and-drop application assembly environment.

Depending upon the underlying, existing application, however, some measure of hand-coding is often inevitable. Low-code platforms must therefore provide sufficiently powerful coding capabilities to handle such outliers – and once developers have created such integrations, the platform should support the ability to place them in a catalog for future use and configuration.

Depending upon the underlying, existing application, some measure of hand-coding is often inevitable. Low-code platforms must therefore provide sufficiently powerful coding capabilities to handle such outliers.



Reducing Application Backlog with Low-Code

As organizations proceed with their digital transformation initiatives, they typically need to build and deploy more applications – and do so faster than before. If application development teams are still following hand-coding practices, however, the wish list for new apps grows longer as the app dev team falls further behind.

We call this list of required but undelivered apps the application backlog – and such backlogs can represent substantial lost business for the organizations that struggle with them.



Fortunately, low-code platforms can help organizations shorten or even eliminate their application backlogs. Because low-code platforms focus on giving professional developers the ability to build applications more quickly while minimizing the need to hand-code, LoB stakeholders can shoulder a portion of the backlog burden.

Empowering business stakeholders to build some apps while professional developers leverage low-code platforms to develop more complex applications offers two primary benefits.

First, since stakeholders are building applications for themselves, those apps are more likely to align with customer needs. Secondly, shifting the burden of application creation from professional developers to such 'citizen developers' frees up scant pro resources to work on more technical tasks.

The collaboration between citizen developers and IT, in fact, is a critical part of this story, one example of the cross-organizational collaboration at the heart of every successful digital transformation.

Such collaboration, however, depends upon the capabilities of the low-code platforms themselves as the features, sophistication, and agility of such platforms increase. The advances in ease of use, integration, and deployment by citizen and professional developers alike has given organizations a practical, cost-effective way to reduce their application backlogs.

It's important to note that no improvement in traditional hand-coding software development methodologies will resolve the worsening backlog problem – only collaboration across the organization combined with broad deployment of low-code platforms will enable organizations to tackle this perennial problem.

DevOps and Low-Code for the Enterprise

As the broader context of enterprise IT has been shifting to an entirely new hybrid IT model, centered on the cloud while incorporating on-premises assets, so too has the organizational context for software in the enterprise. This organizational trend is DevOps.

There are, in fact, two sides to DevOps: first, the organizational transformation as technical teams learn better ways to collaborate in order to deliver and run better software faster than previously possible.



The second: a broad set of tooling that automates many of the tasks that app dev and ops teams must conduct as they create and deploy software.

Much of the discussions around DevOps center on hand-coded software – but in fact, low-code platforms fit squarely into the DevOps tooling environment as part of the continuous integration and delivery principles of DevOps.

As these CI/CD principles become a part of the fabric of how IT delivers value to the business in this hybrid IT context, low-code is rising to the CI/CD challenge. In fact, this world of dynamic, modular software running as workloads in hybrid IT environments is the perfect breeding ground for enterprise low-code.

Enterprise low-code, coupled with architectural, DevOps, and Lean best practices, can finally empower even the stodgiest of big companies to better meet the ever-changing needs and desires of their customers – as long as such organizations can scale up their DevOps efforts without slowing them down.

Enterprise low-code, coupled with architectural, DevOps, and Lean best practices, can finally empower even the stodgiest of big companies to better meet the ever-changing needs and desires of their customers.



Low-code can help with this scaling challenge. With a low-code platform like [Kony Quantum](#), IT organizations are able to establish and enforce an application delivery cadence – a coordinated, iterative pattern of application releases that maintain consistency across the user experience, application capabilities, and the underlying deployment infrastructure.

The result is a modern update to traditional portfolio management, where a governance team can manage a large number of applications across many parts of the business, efficiently allocating resources while managing risk.

However, unlike traditional approaches to portfolio management that slow everything down with paperwork-laden bureaucratic processes, the modern approach is Lean,



lightweight, iterative, and customer-focused. Low-code platforms are essential to achieving these goals of such modern software-driven environments.

Low-Code and Legacy Modernization

As with enterprise integration, the emergence of low-code development platforms drives a hybrid approach to modernization as well.

How does a low-code development team modernize a monolithic legacy application? By targeting specific functionality changes that end-customers require and replacing that functionality with modular software.

Modularity has been a software best practice for decades, and the modern implementation of software modularity is microservices in containerized deployment environments. It's straightforward to implement microservices in a low-code manner, in conjunction with the proper cloud-native architecture that prevents unmanageable interdependencies, exploding management overhead, and increasing brittleness.

Furthermore, companies realize that they can implement a low-code platform that enables them to rapidly prototype solutions and reduce development time, both for greenfield apps as well as updates to existing ones. This capability, in turn, makes it easier to engage with customers earlier and more often during the development process.

Companies realize that they can implement a low-code platform that enables them to rapidly prototype solutions and reduce development time, both for greenfield apps as well as updates to existing ones.



Moreover, the ability to rapidly adapt and change applications and to develop them from a business process perspective allows both rapid iteration as well as the ability to interact with non-technical users for direction and guidance.

Finally, the use of low-code development platforms enables organizations to more closely partner with technology providers to continually adapt the applications as



business needs change over time — helping to ensure that they will not need another round of modernization in the future.

The Intellyx Take: Success with Low-Code

You can think of the enterprise application development challenge as having an 'easy' end and a 'hard' end – the easy end being the front-end, user-facing part where the visible portion of business applications reside, and the hard end being the back-end, where modernization and integration challenges tend to complicate application deployment and slow everything down.

Over the years, low-code tools have focused primary on the easy front-end. Building user interfaces following declarative, drag-and-drop principles is relatively straightforward, and thus many vendors have tossed their hat into the low-code ring, offering various GUI builders to citizen and professional developers alike.

The modern enterprise context for application development, however, goes well beyond the front-end. True, digital efforts require a solid user experience, but customers always require bulletproof end-to-end functionality and performance – and thus the application back-end is every bit as important as the user interface.

The keys to enterprise low-code success, therefore, are a combination of multiple factors. Organizations must avoid shadow IT issues by facilitating effective cooperation among LoBs, the app dev team, and the rest of the IT organization.

Enterprises must place low-code efforts into the context of existing applications, both to integrate with them and, when appropriate, lower the risks inherent in modernizing them.

Perhaps most importantly, enterprise low-code must become an integral part of any enterprise DevOps effort – both as a set of automation tools for accelerating development, integration, and deployment, but also as a framework for greater collaboration as enterprises leverage modern applications as they compete in the digital era.



About the Author: Jason Bloomberg

Jason Bloomberg is a leading IT industry analyst, author, keynote speaker, and globally recognized expert on multiple disruptive trends in enterprise technology and digital transformation.

He is founder and president of Digital Transformation analyst firm Intellyx. He is ranked #5 on Onalytica's list of top Digital Transformation influencers for 2018 and #15 on Jax's list of top DevOps influencers for 2017, the only person to appear on both lists.



Mr. Bloomberg is the author or coauthor of four books, including *The Agile Architecture Revolution* (Wiley, 2013).

About Kony, Inc.

Kony is a fast-growing leader in digital experience development platforms and the emerging low-code platform market; and a recognized leader in digital banking. [Kony Quantum](#) provides low-code without limits, a next-generation low-code app development platform that delivers rich digital experiences. [Kony DBX](#) is the banking and financial services arm of Kony, Inc. and is a globally recognized leader in digital banking transformation. With a portfolio of modern, frictionless applications powered by the industry's most recognized platform, Kony DBX enables banks and credit unions of any size to accelerate innovation — without compromising what's critical.

For more information, please visit www.kony.com. Connect with Kony on [Twitter](#), [Facebook](#), [LinkedIn](#) and [Instagram](#).

Copyright © [Intellyx LLC](#). Kony is an Intellyx client. At the time of writing, none of the other organizations mentioned in this article are Intellyx clients. Intellyx retains full editorial control over the content of this paper. Image credits: [Cindy Shebley](#), [Antonio](#), [Cyril Ribault](#), [Wesley Fryer](#), and [Andre Hofmeister](#).