White Paper

How to Avoid the Low-Code 'Hole'

Jason Bloomberg

President, Intellyx

December 2021

Low-code tools and platforms have taken the enterprise application development community by storm. They promise a simpler, more stakeholder-focused approach to building applications that free developers from routine plumbing work so they can focus on delivering business value.

Low-code platforms allow for hand-coding when absolutely necessary – while simultaneously doing what they can to limit the situations were a low-code developer would ever have to code anything by hand.

There are three primary reasons why a low-code developer might want to customize a low-code application by writing code: in order to build a custom user interface widget, to deal with a complex algorithm beyond the capabilities of the low-code business logic tooling, or to create a custom integration to an application or data source when there's no existing connector that would do the job.

Hand-coding, after all, is an important part of the low-code way of building applications – but the hand-coding option in low-code is not necessarily an improvement. In fact, adding even a few lines of code to a low-code application can cause all manner of problems – a situation we call the low-code 'hole.'

It doesn't matter how smooth the low-code path is everywhere else if you fall into the hole. Here's how to avoid this risky pitfall.

Why Hand-Code at All?

There are three primary reasons why a low-code developer might want to customize a low-code application by writing code: in order to build a custom user interface widget, to deal with a complex algorithm beyond the capabilities of the low-code business logic tooling, or to create a custom integration to an application or data source when there's no existing connector that would do the job.

©2021 Intellyx LLC. https://intellyx.com

Regardless of its purpose, adding these additional lines of code opens the Pandora's box of custom application development, releasing all manner of ills into the world, including security vulnerabilities, testing issues, and increased technical debt.

Following the guardrails that any low-code platform provides to developers mitigates these demons, but also constrains developers who require the freedom that hand-coding can offer.

By jumping into the low-code hole, developers can take all manner of shortcuts, writing code that may meet the requirements of the moment, but with hidden flaws or limitations that will lead to greater quality and flexibility problems down the road.

By jumping into the low-code hole, developers are freed from such constraints. They can take all manner of shortcuts, writing code that may meet the requirements of the moment, but with hidden flaws or limitations that will lead to greater quality and flexibility problems down the road.

Not Agile Enough

This lack of constraints in the low-code hole can also lead to misunderstandings and miscommunicated requirements, especially for custom integrations.

Taking an iterative approach that brings stakeholders and developers together to improve the communication of requirements is a fundamental Agile approach.

However, custom integrations in an Agile manner actually digs the low-code hole a bit deeper, for two reasons: first, it's particularly difficult for stakeholders to accurately communicate what they require from a custom integration, and second, the development work necessary to show a stakeholder what they asked for can be difficult and time-consuming.

In other words, even though hand-coding while building applications with low-code should be rare, the challenges with custom integrations in particular can slow down the entire effort.

©2021 Intellyx LLC. https://intellyx.com

Given how important such integrations are for the overall behavior of the application, there is a significant risk that the entire effort can bog down in the low-code hole.

Krista's Conversational Approach to Custom Integration

Krista Software offers Krista, an iconoclastic alternative to traditional low-code application development.

With Krista, non-technical businesspeople or analysts interact with the platform by telling it what they want. Krista then interacts with the person asking for information in order to hammer out the specific capabilities they require.

Just as with low-code platforms, however, there are certain connections that this conversational approach cannot build on its own. At that point, software developers can use Krista's software development kit to write custom connections that are not already resident in its catalog.

Krista has the ability to mock up the functionality of the back-end systems that application creators want to connect to, enabling them to work with stakeholders to hammer out the specific requirements for the integrations before calling upon the developers to implement them.

Custom integrations are predictably one of the primary examples of this need. However, unlike low-code platforms, Krista handles custom integrations very differently from typical low-code platforms.

Krista has the ability to mock up the functionality of the back-end systems that application creators want to connect to, enabling them to work with stakeholders to hammer out the specific requirements for the integrations before calling upon the developers to implement them.

Once this mocked up integration is complete, application builders hand it off to developers, who code the necessary logic to implement the simulated behavior and appearance. Stakeholders are far more likely to find the result amenable than with the awkward approach that low-code developers must take.

©2021 Intellyx LLC. https://intellyx.com

In other words, the application team is still taking an iterative, Agile approach to hammering out specific functionality when using Krista – only now, those iterations take days instead of weeks as they don't require the developers to step in and code anything until the requirements are complete.

Avoiding the Low-Code Hole

Rapid, lightweight iterations like the ones Krista enables are helpful for distilling all sorts of requirements, but are absolutely essential for properly specifying the behavior of custom integrations.

After all, business stakeholders can be very particular about which data they require, when and where they require them, and how it looks when they finally see them.

Traditional low-code platforms are unable to address these challenges, because custom integrations fall into the hand-coding low-code hole. By using Krista's conversational approach, in contrast, application creators can implement custom integrations without falling into the low-code hole.

The Intellyx Take

Speeding up custom integrations isn't just about going faster (and hence saving money). It's also about getting more done.

After all, custom integrations are typically the bottleneck for most low-code applications – just as they are for traditional software development projects.

Once we resolve this bottleneck, application creation teams can feel free to implement more custom integrations as per requirements in the amount of time it took to deal with a single one in the past.

Krista's mock-based iterative approach to streamlining custom integration also lowers the risk for complex integrations in general.

Without such streamlining, development teams are loath to build anything that requires a complex integration, because so much can go wrong.

If less can go wrong, therefore, the overall risk inherent in such applications goes down, and thus the development team is more likely to take on heretofore risky endeavors. End-users end up the winner, of course – and the business does as well.

©2021 Intellyx LLC. https://intellyx.com

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 Thinkers360's Top 50 Global Thought Leaders and Influencers on Cloud Computing for 2020, among the top low-code analysts on the Influencer50 Low-Code50 Study for 2019, #5 on Onalytica's list of top Digital Transformation influencers for 2018, and #15 on Jax's list of top DevOps influencers for 2017.

Mr. Bloomberg is the author or coauthor of five books, including *Low-Code for Dummies*, published in October 2019.

About Krista Software

Krista Software is in an unrelenting pursuit to help businesses find the right answers. Krista Software produces Krista, a modern Intelligent Automation platform. Krista empowers businesses to leverage existing IT assets by building low-cost automation applications. For more information visit kristasoft.com.

Copyright © Intellyx LLC. Krista Software is an Intellyx customer. None of the other organizations mentioned in this paper is an Intellyx customer. Intellyx retains final editorial control of this paper. Image credits: Kurayba, KiwiDandy, and Alachua County.