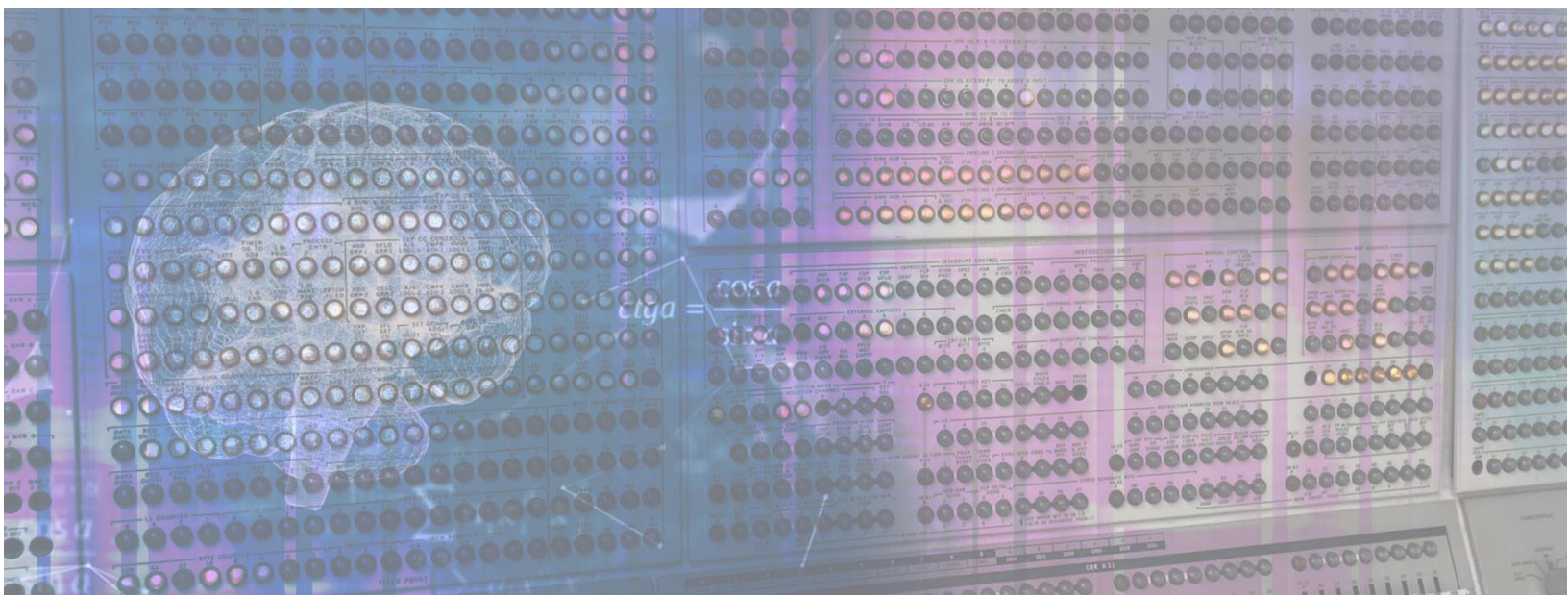




Three Ways AI and Low-Code Application Development Work Together

*An Intellyx Whitepaper for SAP
by Jason Bloomberg, Managing Partner
August 2023*



Whitepaper: Three Ways AI and Low-Code Application Development Work Together

Low-code tools and platforms have reached a level of maturity in the application construction marketplace. Organizations are now depending on such platforms to reduce the cost and time of application development while improving collaboration among developers and business stakeholders.

Simultaneously, the artificial intelligence (AI) marketplace is experiencing a period of rapid disruption, as generative AI platforms like ChatGPT have whipped software vendors and customers alike into an AI-driven frenzy.

Generative AI, in fact, promises to disrupt the low-code space, as it can generate somewhat serviceable source code from nothing but a verbal prompt. Will generative AI replace low-code?

Disruption to such an extent is unlikely. More likely: incorporating AI into existing low-code platforms, augmenting low-code application construction with AI, and applying low-code principles to the difficult work of building, training, and optimizing AI models and their associated training data sets.



AI and Low-Code: Friends or Enemies?

More than any other technology, low-code tools have chiseled away at the proverbial business/IT divide, giving business users more control and input into their software applications while simultaneously empowering professional developers to be more efficient and customer-focused with their software efforts.

Low-code (and its cousin no-code) remove the grunt work of software development off engineers' plates, enabling them to work hand-in-hand with business stakeholders to craft digital offerings with little to no hand-coding required.



Low-code (and its cousin no-code) remove the grunt work of software development off engineers' plates, enabling them to work hand-in-hand with business stakeholders to craft digital offerings with little to no hand-coding required.

At the same time, these tools empower 'citizen developers' from lines of business to try their hand at application construction, putting together apps of varying levels of sophistication, only calling upon IT for trickier challenges like testing, integration, and governance.

While there is variation among the functionality of the various low-code tools on the market, such tools generally provide visual interfaces for application builders, typically with drag-and-drop functionality combined with simple wizards for configuring the apps.



Whitepaper: Three Ways AI and Low-Code Application Development Work Together

Under the covers, low-code tools follow two basic patterns.

In one camp are the code generators – the application builder pushes a button in the low-code interface, and out pops a running application made up of modern source code that can run in whatever execution environment (on-prem or in the cloud) the builder prefers.

In the other camp are the declarative platforms that interpret the visual application in the low-code interface dynamically, automatically running it on the underlying platform (typically in the cloud).

Such was the general state of affairs in the low-code marketplace until early 2023, when large language model (LLM)-based generative artificial intelligence (AI) platforms like ChatGPT arrived on the scene – and suddenly, everything changed.



It didn't take long for people to realize that generative AI platforms could generate source code from simple prompts. The resulting applications aren't perfect but can provide dramatic time savings vs. hand-coding – and all the application builder must do is describe the application they want in sufficient detail.

It didn't take long for people to realize that generative AI platforms could generate source code from simple prompts. The resulting applications aren't perfect but can provide dramatic time savings vs. hand-coding – and all the application builder must



do is describe the application they want in sufficient detail in English (or their human language of choice).

Given the power of today's generative AI and the rapid innovation curve they seem to be climbing, one might wonder if the entire low-code landscape will succumb to this AI revolution.

The truth, however, is more complex. The relationships between AI and low-code are indeed evolving quickly, but there are already several clear innovation trends toward a variety of synergies between these two disruptive technologies.

Innovation #1: Leveraging low-code to include AI capabilities into applications and processes

Generative AI may be garnering most of the attention today, but in reality, AI is an umbrella term that covers many technologies, including machine learning, deep learning, natural language processing, and several others, many of which have been around for years.

All these technologies continue to experience rapid innovation. In addition, dramatic levels of hype, speculation, and occasional malfeasance muddy the waters, making it difficult for decision makers to craft sound AI strategies.

Nevertheless, there are many different AI capabilities that are both mature and reliable, including image recognition, optical character recognition (OCR), document processing, natural language translation, chatbots, and more.

For an application builder leveraging low-code, such capabilities typically appear as modular application building blocks that the app builder can drop into an application.

For example, a mortgage processing application might include an OCR step, taking a pdf of a photographed document and converting it into readable and searchable text. Another common role for AI within a low-code app would be a natural



Whitepaper: Three Ways AI and Low-Code Application Development Work Together

language translation step – similar to using Google Translate but built into the workflow of the application.

AI can also help with the decision making inherent in an application. For example, machine learning is particularly useful for uncovering anomalies in data sets, from performance issues on IT systems to potential fraud in financial services workflows.

Based upon the identification of an anomaly, an application can follow through with a remediation workflow or other process as per the logic of the low-code application.

Innovation #2: Using AI to augment low-code application development

Incorporating AI into workflows or applications is only one facet of the low-code AI opportunity. Perhaps the more powerful innovation is leveraging AI to assist with the application development process itself.



'Next best action' tooling is essentially an autocomplete for low-code workflows. Based upon historical data, for example, AI can predict that the next step in an ecommerce process is to ask for credit card information.

'Next best action' tooling has been available for a few years now. This AI-based capability is essentially an autocomplete for low-code workflows. Based upon historical data, for example, AI can predict that the next step in an ecommerce process is to ask for credit card information.



Whitepaper: Three Ways AI and Low-Code Application Development Work Together

This autocomplete-like capability can work with data as well, automatically formatting data outputs, recommending the appropriate graphs and charts, and cleaning the data themselves.

For example, sales reports tend to have common elements and formats. If a low-code app needs to generate a sales report, AI can format it based upon similar reports in the past.

The code generation capabilities of today's generative AI platforms also follows this low-code pattern – either for application components or for an entire application.

As its name suggests, low-code does allow for some hand coding, typically for specific tasks that require special attention, like custom user interface widgets, integrations to legacy systems, or specialized business logic.

Generative AI is chipping away at all these use cases, generating working code that low-code application builders can incorporate into their applications directly.

As generative AI matures, it promises to enable an entirely new low-code modality.

Today, most low-code tools either follow a 'boxes and lines' flowchart-based modality or a configuration-centric wizard modality (or a combination of the two).

Generative AI adds a third modality, where the application builder prompts the AI with a sufficiently detailed description of the application, and the AI builds to suit – not simply by generating source code, but by creating the configuration-based representation a declarative low-code platform can interpret directly.

Many applications suited to low-code today fall into various categories: ecommerce, project management, registration lists, timesheets, etc. As a result, low-code platforms on the market today typically offer numerous templates for each of these familiar application use cases that developers can modify to suit.

Generative AI takes the idea of application templates one step further, supporting customizations or other unique application requirements that templates alone do



not offer, thus reducing the time and effort to tweak the templates to meet specific business needs.

Innovation #3: Leveraging low-code principles to craft, iterate, and deploy AI models

There is a broader paradigm shift taking place than the two innovations above would suggest. In reality, AI is changing the fundamental nature of an application.

Instead of depending on code or declarative, configuration-based logic, AI applications depend upon *models*. AI models are tools and algorithms that AI specialists use to train computers to process and analyze data in large data sets.

The behavior of the resulting application, therefore, depends upon the models' training as well as the underlying data sets – but not on the result of any human effort to construct the behavior of the application directly.

Instead, the role of humans shifts away from any traditional notion of building an application to adjusting the models, the data sets, and how the data sets train the models.

Today, these activities are difficult and require specialized expertise from data scientists or other AI specialists.

The problem begins with the data. Organizations want to extract value from their existing data, but assembling such information so that models can access and comprehend it is a difficult problem.

Most information in today's organizations is unstructured, consisting of data in pdf files, emails, word processor documents, business forms, and other human-generated artifacts. Traditionally, humans must manually tag many data elements in such artifacts before the AI models can make sense of them.

Low-code tools can dramatically simplify these tagging and general data cleanup challenges by providing point-and-click capabilities for cleaning, cropping,



Whitepaper: Three Ways AI and Low-Code Application Development Work Together

structuring, and tagging data. In fact, low-code tools can also help automate the tagging process while still involving human error correction.

Furthermore, AI model optimization is inherently trial and error-based, requiring multiple iterations. Such trial and error typically continues after deployment of the AI application as requirements evolve and additional source data becomes available.

This ongoing model optimization challenge is a perfect opportunity for next-generation low-code platforms. Where first-generation low-code mitigated the complexity and expertise requirements of hand-coding, we can expect a new generation of low-code tools that empower a broader number of AI application creators to build and optimize AI models and their associated training data sets.



Where first-generation low-code mitigated the complexity and expertise requirements of hand-coding, we can expect a new generation of low-code tools that empower a broader number of AI application creators to build and optimize AI models and their associated training data sets.

Applying low-code to model creation and optimization promises to drive the same human benefits that low-code provided to application development. With low-code, experienced people can work more quickly with fewer mistakes, thus competing projects sooner thus freeing up time to move onto other projects.



Whitepaper: Three Ways AI and Low-Code Application Development Work Together

Meanwhile, low-code also opens up these activities to less experienced people, giving data analysts and other technically-oriented business specialists and subject matter experts the ability to craft and mature their own AI-based applications.

This vision for low-code based AI model building is still largely in the future, but there are important examples of vendors bringing such tools to market today. These cutting-edge products typically fall within specific application areas like fraud detection, IoT-based inferencing, and other specialized arenas.

Given the core collaboration, ease of use, and rapid time to market benefits of low-code, however, the writing is on the wall: the combination of low-code and AI will soon bring AI-based application construction to the masses.

The Intellyx Take

Both AI and low-code are experiencing a period of rapid innovation. Therefore, expect this story to evolve quickly, especially as generative AI in particular reaches higher levels of maturity.

As a market category, low-code had been reaching a maturity plateau – until generative AI dropped a bomb into the marketplace (as it has across so many market segments). The promise of this disruption is rapid innovation, as vendors compete to put together the best AI-driven low-code products they can.

Some enterprises are well-positioned to leverage this disruption, as early adopters jump on new technologies, taking the inherent risks to get the jump on their competition. The majority of companies, however, are best suited to wait for the early adopters to shake out the market.

There is no shame taking the less risky road if the risks of early adoption don't suit your business. Leverage the mature low-code products on the market today while ramping up your AI strategy. When AI-empowered low-code matures, there will be plenty of opportunities to take advantage of such platforms.



Whitepaper: Three Ways AI and Low-Code
Application Development Work Together

About the Author



Jason Bloomberg is the founder and managing partner of enterprise IT industry analysis firm Intellyx. He 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 #13 on the [Top 50 Global Thought Leaders on Cloud Computing 2023](#) and #10 on the [Top 50 Global Thought Leaders on Mobility 2023](#), both by Thinkers 360. He is a leading social amplifier in Onalytica's [Who's Who in Cloud?](#) for 2022 and a [Top 50 Agile Leaders of 2022](#) by Team leadersHum.

Mr. Bloomberg is the author or coauthor of five books, including [Low-Code for Dummies](#), published in October 2019.

About SAP Build

SAP Build provides everything you need for low-code application development and automation. Create enterprise apps, automate processes, and design business sites with drag-and-drop simplicity. With SAP Build you can build visually, integrate all your enterprise assets seamlessly, and collaborate securely across IT and business with modern application development. To learn more and experience SAP Build for free go to [www.SAP.com/Build](https://www.sap.com/Build).

Copyright © Intellyx LLC. SAP is an Intellyx customer. Intellyx retains final editorial control of this paper. No AI was used to write this paper. Image credits: autogenerated at [Craiyon](#).