The Importance of Low-Code Developer Tools to Hyperautomation Technologies

January 2022

Questions posed by: ServiceNow
Answers by: Arnal Dayaratna, Research Vice President, Software Development

Q. What is hyperautomation in a business context, and why is it important?

A. Hyperautomation in a business context is the amplified acceleration of automation into business and operational processes. Whereas the implementation of automation has historically included the use of discrete technologies such as DevOps, artificial intelligence/machine learning (AI/ML), robotic process automation (RPA), and low code for specific tasks, hyperautomation refers to the business strategy of automating as many business processes as possible by using a multitude of technologies in parallel and in cooperation. As such, hyperautomation is defined by the orchestration of a multitude of technologies, platforms, and infrastructures to maximize the operational efficiency of work performed by employees and business and technology processes.

Hyperautomation is important because it involves the orchestrated use of process-agnostic technologies that can be used across a multitude of organizations, lines of business, departments, and use cases. Organizations are increasingly using coordinated assemblages of automation technologies as opposed to discrete and localized implementations of automation technology. Put differently, hyperautomation entails the intentional aggregation of discrete automation technologies that can be used for a variety of use cases and tasks within an organization. The ability to derive holistic insights and continual improvement from the end-to-end view of tasks and processes creates previously unforeseen business value. The linking of the technologies together, which also links people and processes, creates a more intelligent approach to automation.

Benefits of hyperautomation include increased operational excellence, cost savings, improved productivity, expanded collaboration, enhancement of the accuracy and relevance of data-driven insights, increased compliance, and augmented risk management. Another key benefit is improved employee satisfaction resulting from the automation of manual processes, thereby freeing up employees to focus on higher-value work. Beyond that, the ability to democratize automation for users of any skill set is a new frontier not just for employee experience but also for employee business impact and innovation.
Q. How do low-code technologies fit within the landscape of hyperautomation tools?

A. Low-code technologies constitute the secret sauce underlying all hyperautomation technologies given that AI/ML, DevOps tooling, RPA, intelligent process automation (IPA), event-driven software architecture, and integration platform as a service leverage low-code tooling for end users. Hyperautomation technologies are deepening their usage of low code as a means of accelerating the velocity with which digital solutions can be created and empowering business users to develop solutions in conjunction with the business. As a result, low-code technologies are illustrative of the direction of hyperautomation technologies because of their ability to enrich and accelerate the development of digital solutions that specialize in automation.

Another way to think of low-code technologies is that they constitute the foundational infrastructure on which hyperautomation technologies are rendered available to end users. The ability to use low code to automate the production of a digital solution allows business users to rapidly create digital solutions for pressing business problems. This means that low-code technologies democratize development by rendering the development experience accessible to business technologists who lack experience with custom coding and scripting. This expansion of the universe of resources that can participate in software development enables organizations to develop applications more quickly. Additionally, the democratization of development enabled by low-code technologies increases opportunities for employees to collaborate on the development of digital solutions. This enhancement of collaboration provides an opportunity for the automation-specific needs of more workers to be heard and recognized.

Q. What are some of the key challenges of adopting low-code development tools within an organization?

A. Organizations that are adopting low-code development tools are challenged with providing developers with guidance about which use cases and types of applications are optimal for low-code developer tools. As low-code tooling becomes more powerful, employees can use low code to create more sophisticated and complex digital solutions. Business professionals who tackle the challenge of creating more complex applications using low code run the risk of creating poorly designed applications that experience various performance-related challenges. To mitigate this risk, IT-based developers must collaborate with business technologists. IT needs to provide line-of-business (LOB) developers with guidance about best practices for low-code development. Best practices are especially important to ensure consistency and standardization in the development process as a means of creating repeatable and recognizable patterns for application development.

Moreover, IT has to create guardrails to ensure that employees do not create digital solutions that have deleterious consequences for organizations. IT will need to review and update such guardrails periodically to ensure that they accomplish the goal of mitigating risks associated with development without excessively constraining the creativity and initiative of LOB developers.
Another challenge is the perception among full-time developers that low-code development tools are incapable of creating production-grade, mission-critical applications. Some developers are likely to resist adoption of such tools based on the belief that command-line coding tools are uniquely capable of developing production-grade applications. Additionally, developers may feel that digital solutions developed using low-code developer tools are not sufficiently extensible or lack the depth of sophistication that is characteristic of command-line coding developer tools. That said, professional developers are likely to revise this perception and adopt the view that low-code developer tools expand their range of choices for development and accelerate the development process after they acquire more knowledge of tools within this landscape.

**Q. How should organizations prepare to address and manage the key challenges associated with the adoption of low-code development tools?**

**A.** Organizations can provide developers with guidance about which use cases are optimal for low-code developer tools. They can do this by developing an adoption strategy that features a roadmap for the progressive adoption of low-code developer tools into application development initiatives. Such a strategy should specify goals for the adoption of low-code developer tools among full-time developers as well as their use for the development of both customer-facing and internal mission-critical, production-grade applications. Organizations should delineate an annual percentage share of their application development portfolio that will be completed using low-code development tools that includes details of how that percentage is forecast to increase annually over the next five years. Additionally, organizations would do well to provide IT leaders with a set of criteria that can be used to evaluate whether a development project falls best within the purview of low-code development tools or pro-code integrated development environments (IDEs).

Meanwhile, organizations can address the perceptions of low-code development tools among developers by illustrating the expanded range of development options available to them as a result of such tools. Deepening the exposure of low-code professional developers to low-code developer tooling is likely to underscore the sophistication and value of the technology for a multitude of development use cases. Given that professional developers are tasked with shipping code faster without compromising security or quality, their enhanced experience with low-code development tools gives them the insight to accelerate development while enabling them to create richly nuanced functionality.

Further, organizations can create a low-code center of excellence that takes responsibility for creating and refining best practices for developing, testing, deploying, monitoring, and managing low-code applications. This center of excellence should also centralize training-related resources and documentation and render them easily accessible to interested parties. Moreover, organizations would do well to implement governance protocols that provide guidance regarding application design, data security, testing, monitoring, role-based access control, and ongoing operational management.
Q. What are some of the best practices organizations should follow to facilitate collaboration between enterprise IT-based developers and LOB low-code developers?

A. Organizations should create structured pathways that expedite and enrich communication between developers. For example, one method of fostering collaboration between IT-based and LOB low-code developers involves leveraging collaboration functionality specific to low-code developer tools that empowers a multitude of developers to work together. Such collaboration tooling can be used to enable regular conversations between IT developers and their LOB counterparts. A regular cadence of conversations between IT-based and LOB low-code developers is essential to providing IT-based developers with insight into the problems faced by LOB developers. This insight can be used to develop best practices and training collateral that are tailored to the specific needs of the LOB developers at their organization.

Another way for organizations to deepen collaboration is to create processes whereby enterprise developers can outsource development-related work to LOB colleagues. The development of such processes would require enterprise IT-based developers to identify development workstreams that can be reliably delegated to LOB low-code developers. Such collaboration deepens the skills of LOB developers while enabling enterprise IT developers to focus on high-value development tasks. This practice of transitioning development-related work from enterprise IT-based developers to LOB low-code developers is best accomplished within the purview of an LOB development program that continuously assesses the skills and availability of LOB developers.

About the Analyst

Arnal Dayaratna, Research Vice President, Software Development

Dr. Arnal Dayaratna is Research Vice President, Software Development at IDC. Arnal focuses on software developer demographics, trends in programming languages and other application development tools, and the intersection of these development environments and the many emerging technologies that are enabling and driving digital transformation. Arnal’s research examines how the changing nature of software development relates to broader trends in the technology landscape.
MESSAGE FROM THE SPONSOR

Digital transformation is more relevant than ever. Growth and success require faster, more agile automation delivered through software. That's where low-code automation solutions come in. Powered by the Now Platform, ServiceNow App Engine fuels rapid delivery of brand new workflows with great experiences for everyone on the platform of platforms. With more people building and automating with less complexity, ServiceNow lets you create low-code apps fast, automate any tasks across systems, and safely scale cross-enterprise experiences that users love. Learn more.