How Low-Code/No-Code is Shaping the Future of Software Development

For decades, the software development process has been considered complex and a long-drawn-out one. The work was carried out by teams of developers who had time to think about how best to build and deliver applications to their customers. How times have changed. Over the last year, the majority of the world has been grappling with working remotely. This is driving more demand than ever to deliver top-quality as-a-service products. Demand is for applications and platforms for consumers and for business users alike easily integrated across multiple systems. They must be delivered faster and at a lower cost than ever. According to a McKinsey report, the global software industry grew at twice the rate of the overall market over the last decade but the software industry's profitability fell by half.

SaaS and PaaS companies are now looking for the best ways to meet the needs of their customers. Low-Code/No-Code development holds great potential to simplify and streamline parts of the software development process. It is not a cure for everything facing the as-a-service software industry and it does also present potential challenges. However, they can be powerful tools when wielded effectively.

Differences between Low-Code versus No-Code

Low-Code refers to a form of developing software, applications, or databases through a graphical user interface (GUI) and can be fully customized with a minuscule amount of programming.

Low-Code requires the use of a visual integrated development environment (IDE) as the user simply has to leverage visual components to assemble their custom application. Low-Code offers the option to view or edit the source code and is cost-effective.

No-Code development, on the other hand, can be carried out by anyone who desires to tailor an application to his/her needs without any programming knowledge. Unlike Low-Code, No-Code does not offer the option of viewing or editing the source code. The No-Code space brings into play 'citizen developers' who build functional but generally limited apps without having to write a line of code. By leveraging a GUI, users can utilize No-Code development platforms by dragging and dropping features straight into their application framework.

However, implementing Low-Code/No-Code development has its share of challenges. Low-Code lacks customization and also has integration issues, especially with legacy systems. More importantly, security and reliability are other concerns using Low-Code as there is a degree of risk involved in working with Low-Code as you don't have complete control or knowledge of the entire coding process. No-Code also has its challenges. It is not always a cheaper option. In fact, at times, it can be even more expensive than outsourcing or nearshoring code development. No-Code also has limitations concerning scalability.

However, no matter the challenges of using Low-Code/No-Code, the potential upside makes them necessary for companies to evaluate their fit for their development efforts. Let us take a deep dive into how the Low-Code/No-Code movement is revolutionizing the software industry.

Fast-Paced Software Development

Organizations leveraging Low-Code/No-Code platforms can enable software development within a few days or weeks, unlike traditional application development that can take weeks or even months. This allows for rapid speed-to-market as well as ensure software updates are carried out regularly as anyone can make changes irrespective of their level of coding knowledge. The absence of code writing also frees up software developers to focus on other more critical development tasks or even to focus on improvements to products or new products altogether – driving potential revenue.

Cross-Platform Compatibility

Custom applications largely developed through traditional methods are built for one platform or another, e.g. Android or iOS. Such platforms have limitations owing to the costs involved in developing two different platforms. Certain Low-Code/No-Code development is cross-platform and can be deployed across multiple devices, but these are limited.

Cost-Reduction

Low-Code/No-Code serves as an enabler for building more applications/platforms in less time, which helps organizations reduce development costs. They also eliminate the need for hiring more software developers, thus helping keep a tight rein on staffing costs and improve productivity.

Lower Maintenance Burden

Software maintenance is a big responsibility for organizations. Low-Code reduces the software maintenance burden by reducing the plumbing work from day-to-day development. Since Low-Code ensures components are standardized, pretested and ready-made, organizations have to deal with fewer bugs and integration issues than in the past. It also ensures software developers spend less time on maintenance.

No Need for Training In-House Talent

Organizations can leverage Low-Code/No-Code to reduce time and effort in training their workforce. Such platforms do not need organizations to have developers trained in different programming languages. Often basic HTML knowledge is more than enough, which paves the way for organizations to deploy in-house IT professionals to execute their software development solutions.

Conclusion

Low-Code/No-Code is emerging as a hot topic for all types of companies. According to a Forrester report, the Low-Code market is poised to touch an annual growth rate of 40%, with spending forecasted to hit a whopping $21.2 billion by 2022. These industry disruptors allow companies to build an application or platform without extensive manual coding. By harnessing the power of Low-Code/No-Code, organizations can transform the software development process. It allows for software/platforms to be built faster and easier than ever before driving new revenue while controlling costs. To put it simply, Low-Code/No-Code is a tool that can be leveraged to develop many new applications or platforms.