



How ISVs can build and customize solutions **faster using low-code**



Challenges ISVs face using **Legacy Systems**

Traditionally, ISVs have built software solutions that needed to be installed on specific computers, configured and administered for use. Time-to-time, this software needs upgrades, which incurs additional costs, time and effort for enterprises to keep their software up-to-date. Apart from this, no matter how good a software package is, it will require modification to fit the way the enterprise works. For this, the business demands the software to be readily customizable for a specific customer.

As mentioned above, customizing traditional software solutions itself requires difficult and expensive custom coding. Imagine what a nightmare it is for ISVs to manage updates to a customizable, configurable and installable software. Even if they want to make it easier for their clients, ISVs find themselves scrambling to support multiple versions of the product. This dramatically increases the cost of development, product management and support.

Besides cost, factors such as integration and changing the UI often discourage organizations from modifying their packaged software. Modifying the UI of a customer-interaction system is a major undertaking. Beyond the change itself, the modification will likely involve planning and design considerations, integration with other applications and QA and testing to ensure the change does what was intended without breaking anything in the process.

The traditional systems' headache:

- The cost factor - to maintain, keep it running and customize to newer needs
- The time factor - to upgrade or modify the software based on specific needs
- The people/ skill factor - to add new features on top of multiple legacy systems

Modernizing legacy systems: Fast forward to cloud era

Current generation software has become cloud-native and serverless, delivered and consumed on the cloud. Subscription based, multi-tenant and continuous delivery on cloud are the advantages which every ISV wants to reap. Software has to go through critical architectural changes to be able to adopt this model which offers microservices architecture, horizontal scalability with containers and multi-tenancy.

Today with cloud computing, every segment of the technology industry has changed dramatically and ISVs are no exception. Not so long ago, buying software involved a high one-off purchase cost for a solution that one could install on local computer or server.

Now, more companies are choosing to use SaaS applications rather than deal with the time and expense of implementing and maintaining on-premises software. Seeing this, ISVs are transitioning from an old model of packaged software to SaaS as it is browser-based and constantly evolving.

Cloud, the emerging technology trend has revolutionized the businesses are run today. Gartner recently said “Cloud computing is mainstream, with approximately 58% of organizations well down the path of using cloud services to support some aspect of their business.”

For many ISVs though, this process is still at an early stage, as transforming legacy on-premise code and existing architectures is expensive and complex.

Let's look at the key challenges that ISVs face while implementing cloud-based solutions:

<p><i>Migration challenge</i></p>	<p><i>Migration requires a technology feasibility which ISVs must look into before proceeding. It may be that legacy tools of the cloud infrastructure may not integrate with the traditional systems. One of the major roadblock for ISVs not wanting to shift to cloud is the higher dependency of existing infrastructure on different cloud applications</i></p>
<p><i>Security landscape</i></p>	<p><i>Security risks of mobile applications concerns many a user. As multitenant cloud environment share the database infrastructure, they pose a greater security risk. Thus cloud adopters must have a proper strategy for protecting their IP address in the face of data breaches and cyber threats</i></p>
<p><i>API Integration</i></p>	<p><i>Although, depending on third party API may reduce development cost, it may result in poor UX in cases of downtime. On the other hand, well-tested APIs deliver robust experience always. Additionally, there is the risk of some of the existing features getting deleted when upgraded to a third-party application</i></p>
<p><i>Automation testing</i></p>	<p><i>Automation testing enables shorter development cycles and delivery of instant feedback. Successful automation involves the right set of skill sets, tools, and approach. But making changes to the existing testing processes involves multivariate data. Moreover, it takes a lot of time to develop a testing framework that synchronizes the business logic with test case executions</i></p>

It is a make or break decision for ISVs moving to cloud. Modernization effort could lead to refactoring of certain portions of the application, possibly re-writing few modules and re-using some parts. It is a huge learning curve and requires a lot of expertise to move from monolithic to a cloud-native solution.

There are several critical decisions which needs to be taken:

- Application stack upgrade
- Relational database vs no SQL
- Creating micro services
- Establishing API contracts
- Session-less architecture for achieving horizontal scalability
- Security and propagating user context across micro services

ISVs understand that they need to re-define their product strategy when the technology landscape is getting disrupted by IoT, cloud computing etc. Some key questions that ISVs must consider are - What technologies should they invest in? How can they reduce the associated risk of failure or costly mistakes? One has to keep in mind that what looks promising today can go off the shelf in less than 6 months. So it's critical to find the right technology before investing time, money and resources.

The customization problem

No matter how good a software package is, every client will have their own customization requirements from the software. They may need to add a few more features or modify the existing ones in order to serve their customers in the best possible way. The client's own IT staff may lack the technical skill, or the knowhow needed to make modifications to the software. In such a case, the solution provider may have to outsource developers to do the job. This may seem a feasible solution for a few clients and few applications. But when catering to a large number of clients running multiple applications requiring customization, this becomes a huge challenge. It is time consuming and dramatically increases the cost of development, management and support.

The multi-tenant model approach of cloud-based software does not leave much room for customization. Multi-tenancy is an architecture in which customers share computing resources in the cloud keeping their own data isolated and invisible from others. In this model, each tenant can customize some parts of the application but not the application's code. Most SaaS solution providers use the multi-tenancy model because it is less expensive and helps to streamline updates. So even though business models of traditional ISVs are becoming extinct, customization still remains a major problem of ISVs providing cloud-based solution.

Getting all of this right requires a huge transformation within the ISV organization, while making the current business function simultaneously using the old software delivery approach. This is a make or break for ISVs, as there is a huge risk involved in this kind of transformation.

So, problems that modernization brings to the table:

- It is a big leap for ISVs to go to the cloud era of software delivery as it requires huge modernization efforts
- They need to customize the software for every customer which is not only tedious but is very restrictive and usually doesn't meet all the customization needs
- Additionally, they need to build a component-based customization model in order to scale to a large customer base

Enabling upgrades to customizable portions of the software is a herculean task, especially when there are code-based customizations which requires this code to be compatible with the rest of the software stack. One way to get out of this customization dilemma is embracing a platform approach for delivering software.

Making way for the **low-code model**

Low-code platforms are next generation development platforms poised to change the app development landscape by embracing all the latest architectural practices along with move to cloud-based delivery.

What low-code platforms promise:

- Rapid development
- Easy integration
- Responsive design
- Scalable architecture

This is what new generation software delivery needs. ISVs can take advantage of low-code platforms for their customization needs as well as building application software solutions adhering to the current day architectural best practices to leverage cloud-based delivery.

Visual drag-n-drop and component-based development approach of low-code platforms allow enterprise users who use ISV software to customize and build newer functionality, as well as extend existing features offered by the ISV software.

The advantage that **WaveMaker brings**

WaveMaker Low-code Platform uses a code-behind approach generating truly open-standards based code using popular frameworks like Angular, Spring, Hibernate, Cordova, etc. It combines the power of ease of development through visual and model-driven approach with extending code using standards-based frameworks.

In addition to bringing an open-standards approach, WaveMaker application architecture leverages micro-services with session-less architecture and auto-generated REST API contracts for backend services. Completely decoupling frontend and backend layers through these APIs, WaveMaker helps develop large-scale, cloud-ready applications. This is a big win for ISVs to take advantage of all the benefits offered by WaveMaker for customization and cloud-enabled delivery.

An ISV application integrated with WaveMaker, will enable an organization's business process people to use a visual, drag and drop tool to change content and customize the process workflow themselves, without the need for full scale application development, integration and testing. The correct screens are automatically built without needing complex hand-coding.

With WaveMaker, change no longer prevents a company from responding quickly to events that impact its service. Its customers are kept up-to-date on the situation almost immediately. The organization saves itself from having to race through a disruptive scramble to respond to business changes.

Contributing to a **Platform ecosystem**

As the ISVs reap the benefits of cloud-based delivery and SaaS based subscription model, the next step is to build a sustainable model for developing components, distributing them to essentially create an ecosystem to thrive. Component-based customization model is a powerful approach where enterprise customers will develop extended components based on their own requirements.

WaveMaker Platform has several components that can be developed independently and use them as part of the application such as Templates, Prefabs or UI Integrated Components, Themes and Project Shells. These reusable components offer great advantage for ISVs and enterprises to collaborate as part of the software solution delivery and this will lead to building a thriving ecosystem.



About WaveMaker

WaveMaker is one of the most open, extensible and flexible low-code platforms that elevates your enterprise application delivery while keeping in mind the requirements of Software Developers, Citizen Developers/Business Users, IT Architects and CIOs.

[Start your free trial](#)