

# How Migrating Legacy Systems Improves Performance & Adaptability

by Joe Carroll

The antiquated business systems, apps, and processes that were revolutionary 30+ years ago can't compete (or integrate) with modern technology. Even if the legacy systems of yore are still in operation and mission-critical, it's very likely that they are monopolizing IT resources while also being incompatible with new software. In other words, legacy systems are holding businesses back.

Sooner or later, every organization that is looking to embrace digital transformation has to consider the best legacy migration and modernization strategy — which is often easier said than done. In this article, we explore the different challenges of legacy migration and how low-code technology can streamline the development of new, custom business apps to replace, supplement, or improve your existing systems.

## Types of Legacy Migration Projects

There are two types of legacy migration projects: lift-and-shift and business transformation.

**Lift-and-shift legacy migrations** are typically initiated by IT because the underlying infrastructure or technology needs to be updated or replaced. This is often for cost or technology obsolescence reasons. In these scenarios, the IT organization is looking to rebuild the existing system on a one-to-one basis, and there is no compelling or urgent need for significant change on the side of the business. They simply want a new system that allows them to continue doing what they've been doing.

Most of the legacy migrations we see our customers undergo are **business transformation initiatives**. Driven by the business, these apps are meant to replace legacy systems that don't sufficiently support business processes or provide the right user experience. While these legacy migration projects require new functionality, they often must support current processes as well.

Within the business transformation category, we see two types of systems:

- **Core legacy migrations:** Larger systems built by central IT, such as portals and custom ERP systems. These apps require scale, performance, and complex legacy database migration.
- **Non-core legacy migrations:** Smaller, departmental apps originally built by the business using tools like Microsoft Access, SharePoint, or Lotus Notes. The defining characteristics for successfully migrating these systems are business enablement, centralized governance, and easy data migration.

# **Legacy System Migration Challenges (and the Low-Code Solutions)**

## **1. Lack of flexibility creates tomorrow's legacy**

Legacy migrations require organizations to build a system that is flexible to able to adapt to changes. The last thing IT wants is for the new system to become tomorrow's legacy.

Low-code platforms enable organizations to employ a modern app architecture that promotes agility by leveraging micro services. Look for a platform that enables the easy creation of autonomous apps and services that can be recombined and shared, while facilitating fast, frequent change cycles so the system can evolve to meet new business needs. It is critical to take advantage of a future-proof platform so you aren't creating an unmaintainable set of apps or piles of machine-generated code.

Enabling developers to rapidly take advantage of best-of-breed modern technology stacks without having the skills to be an expert in all of them will create a high level of flexibility that is fully documented and visually constructed.

## **2. Mission-critical apps fail to deliver at scale**

Many core legacy systems fail to deliver the required performance at scale. Look for a low-code platform that ensures the new solution can be deployed with the required resiliency and high availability for mission-critical use. A low-code platform with a cloud-native architecture enables automatic failover for continuous operation of business-critical apps, ensuring that they don't run into the same performance issues as legacy systems.

## **3. Inadequate oversight of quality**

Due to the size and complexity of most legacy migration solutions, there is often inadequate oversight of the quality of the app. It is important to be able to proactively monitor and address the quality of the applications to prevent technical debt and improve long-term maintainability. Look for a low-code platform that embeds automated quality, testing, and performance monitoring within the development lifecycle.

## **4. Usability is an afterthought**

With this type of application, usability is often an afterthought, blunting ROI. Use low-code development to take a user-centric, design thinking approach when creating the new solution. A deeper understanding of the users and business context can help close process gaps that existed in the legacy system, delivering an end-to-end solution that drives substantial productivity gains.

This level of understanding may also result in the incorporation of new capabilities that weren't available in the legacy system (e.g. mobile, conversational UI), or the removal of unused features. Both help deliver a more focused and engaging user experience. Look for a platform that enables the business to participate in the design process to ensure app usability and success.