mx mendix
Go make it™

eBook

beyond build vs. buy

How to get there > >

Contents

<CHAPTER 01>

Trapped in an Unshifted Paradigm

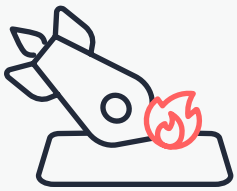
Why We Used to Buy

Every organization in our digital world has to determine whether it is better to buy Commercial Off The Shelf (COTS) software or build software that directly meets their needs. For over 20 years, the argument has strongly favored buy. Historically, COTS has had several advantages: faster time-to-market, lower engineering costs, lower risk, and "immediate" deployment for use.

Building fit-for-purpose, or custom, software gave organizations agility, but was also considered risky and associated with large upfront costs, operational and maintenance expenses, implementation concerns, lengthy time-to-market, and expensive developer resources.

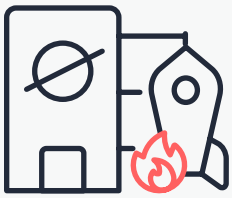
The seminal research survey[1] done in 1994 by the Standish Group showed 31% of build projects were canceled before completion. The survey also indicated 53% of projects cost 189% of their original estimate. **24% of those respondents said the biggest reason why projects failed was due to incomplete or changing requirements and specifications and 12% of respondents said it was lack of user input. Think about that.** Over 36% of respondents indicate a lack of collaboration and real-time feedback as a reason for failure in 1994.

(1994)



31% of build projects were cancelled before completion

(2009)



68% of IT projects were failures from the start

Follow-up research by IAG Consulting[2] in 2009 found that 68% of IT projects were failures from the start—doomed from the beginning by poor requirements. 50% of the unsuccessful projects were considered runaways, all of them flagged as lost hopes due to at least two of the following reasons:

+ The project took nearly twice the estimated time to complete.
+ Costs exceeded more than 1.6x budget.
+ Less than 70% of expected functionality was delivered.

Not only were organizations failing to build, but they could not compete based on the core system choices they made.

This was the era of buy.

<CHAPTER 02>

Dollars, Dollars Everywhere

Paying the Hidden Costs of COTS

In the last decade, the software market has undergone considerable transformations with significant consolidation in the market. Long standing players have disappeared. **Now everyone has the same CRM, ERP and other core systems (i.e. SAP, Teamcenter, Salesforce). Organizations can no longer create durable competitive advantage based on their core system choices.** This is fundamentally different than it was just five years ago and is forcing a change to how we think about building vs. buying.

The fantasy of COTS is that with a few plug-ins, connectors, and bit of coding those solutions will neatly dovetail into your core systems. **The reality is this—COTS software requires extensive customizations, workarounds, and creative rigging to achieve a functional integration with an organization's IT infrastructure.** Implementing a COTS software solution always ends up with people Frankenstein-ing their way to full functionality.

For example, say you buy a product lifecycle management (PLM) system to improve your IT Service Management, but (surprise, surprise) you realize it has some core workflows that don't meet your needs. You think you're about to smoothly integrate a comprehensive solution and when the reality of implementation sets in—you see the glaring need for software that lives in between your core systems. The new PLM system has to integrate with SAP to be able to continue to manage an existing process, which requires more customizations.

In addition, imagine that to manage a long running process the PLM system has to integrate with SAP and so on; you end up building anyway. Ultimately you bought the system assuming you were buying a complete solution out-of-the-box and didn't anticipate the amount of customization (i.e. building) required after the purchase. The problem is that purchased software isn't structured to support complex builds. It isn't good at it.

Generally, purchased solutions get you 60% of the way there, but you have to build to get the other 40%. Since most organizations underestimate the amount of building required, the work is usually rushed and of poor quality.

This happened in 1994, this happened in 2009, and it continues to happen today. Organizations buy a solution only to end up with Shadow Build, thereby negating the expected "savings" regarding costs, time, and resources they expected to garner from a bought solution.

> **"If we are to achieve the expected gains from purchasing software versus building it ourselves, then for the entire life cycle of the products, we cannot allow any modification."**
>
> Cindy Shelton, *Business Process Reengineering with Commercial Off-the Shelf Software*[3]

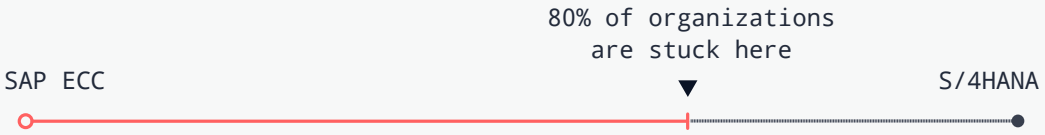
Essentially, even when you buy, you build.

<CHAPTER 03>

Losing Big with Legacy Frankensteins

How Cobbled-Together Core Systems Breakdown

At least 80% of organizations are stuck in their journey of migrating from SAP ECC to S/4HANA because they bought something and customized it beyond recognition. Now they are paralyzed. They can't migrate over unless they scrape off everything they customized. When you buy and then customize, you don't take into consideration the scale and integration issues, and before you know it, you are locked in. This creates a continuum of technical debt that eventually manifests as an urgent business problem.



They decided to buy and then had to customize. Now they are in trouble. Outcomes are rather different for organizations that embrace build from the start.

We are not suggesting that organizations build their core systems. Businesses need to think about their architecture in terms of core systems of authority. In most cases, it is not practical to build your own PLM or Enterprise Resource Planning (ERP) system. The important thing is to understand that you are buying a discreet system of authority and it does important core functions, but everything outside of these core functions should be built to help differentiate your business.

Ultimately, the choice to buy or build centers around the need to extend your differentiation, and to make your organization more efficient so you can compete more effectively. Today's businesses can't afford to lose time or money buying and implementing the same solutions that their competitors are using.

<CHAPTER 04>

Get Clever with the Cutting Edge

Innovative Tech Solves Old Problems

The companies investing in cloud, agile, and DevOps are on a glide path to low-code development, because the convergence of all these technologies and methodologies together have made build a favorable option.

In addition, developments in serverless, platform-as-a-service (PaaS), software-as-a-service (SaaS), and low code give businesses the best of both worlds and enable competitive advantage across the whole landscape. Whether it's advantage in core system agility, or systems of differentiation and innovation, combining these technologies allows you to optimize your processes while mitigating all the concerns that were once associated with building custom software.

Businesses can now embrace the option to build for the first time for two reasons: **one, because now your core systems are open and have rich interfaces**, so you can build software that lives in between them and build fit for purpose solutions. And **two, because the means of cost-effective, rapid builds have matured so much in the last few years**. Technical capabilities have truly evolved, so old prejudices against building are no longer relevant.

The evolution of these tools and technology have also democratized the role of the developer. According to the IAG Consulting research[4] from 2009, 70% of companies stated that the level of competency required to complete projects was higher than that of the assigned developers. The impact of this skills gaps directly increased project time, cost, and risk of failure.

the average developer's competency does not meet their needs

Fast forward to present day, and companies are still facing the same challenges. Market demand for mobile app development services have grown at least five times faster than internal IT organizations' capacity to deliver them[5]. This trend is exacerbated by the lack of enough professional developer talent—with a 1.6% unemployment rate, enterprises can't find good prodevs fast enough, even if they are willing to make a bigger investment in IT. In fact, according to IT world[6], it is not uncommon for a company to spend 8-12 weeks or longer hiring a specialized team of developers.

The solution to the parsity of professional developers isn't an exorbitant IT payroll. The answer is to empower citizen developers throughout your organization to contribute to the building process using the new, more accessible tools now available.

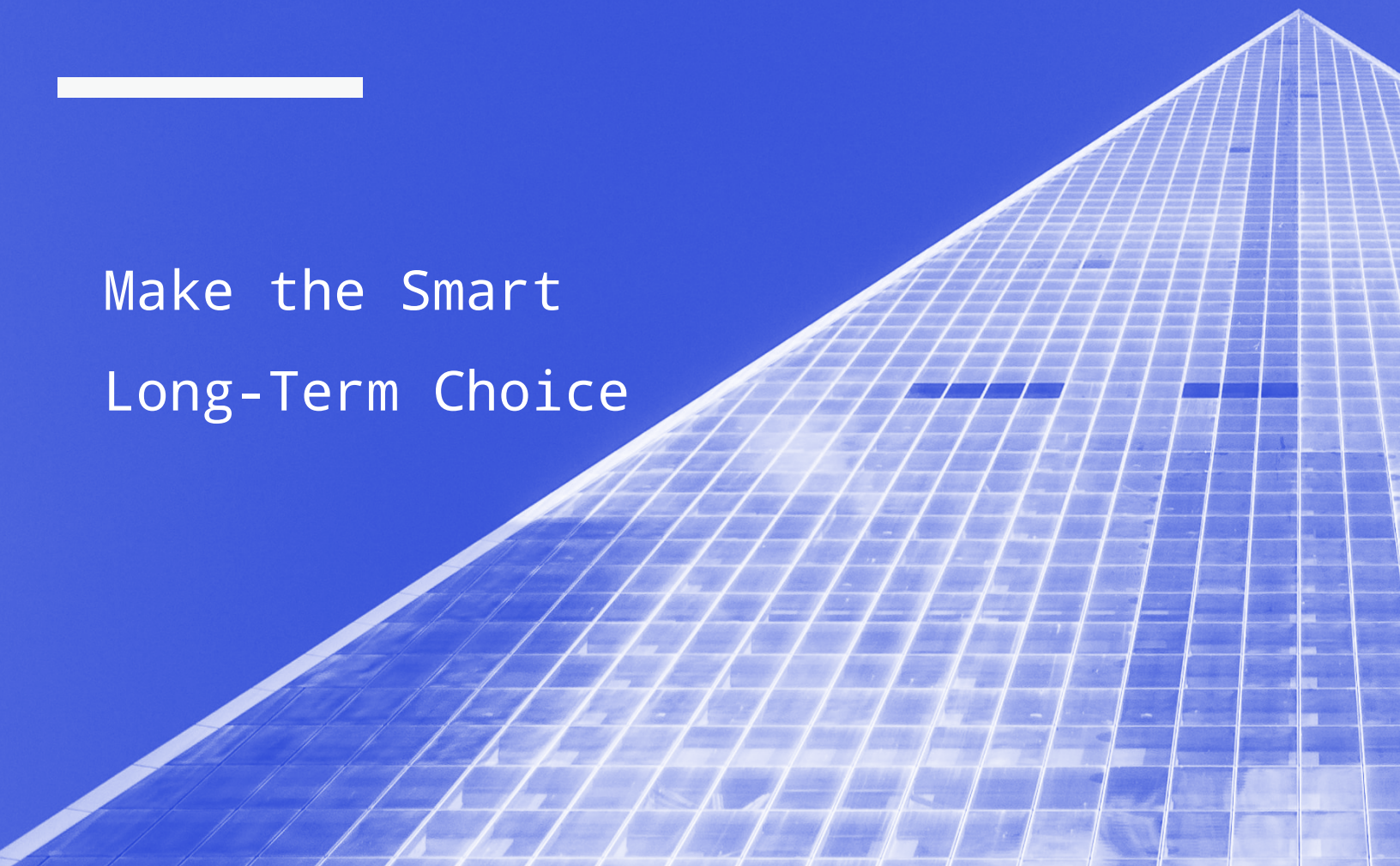
Building is no longer the exclusive purview of the professional developer. New tools and technologies opened the development field to knowledge workers not trained in traditional coding—making building faster, more accessible and more cost-effective.

This new way to develop puts build on the table in a big way.

<CHAPTER 05>

Build for the Win

Make the Smart Long-Term Choice



Recent innovations mitigate the concerns that have long been associated with building and customizing software. Pros + Cons in the conversation around enterprise software acquisition have changed.

Innovative Integration

Bought software doesn't enable easy integration with new technologies like cognitive services, machine learning algorithms, or augmented reality. The right cloud and low-code platforms support integration[7] with these innovative technologies and services through industry-standard interfaces, ensuring that new solutions can effortlessly connect.

Cost Reductions

By developing with these platforms, your organization can host on the vendor's cloud. This allows you to provision services and applications without having to deal with servers, network, routing and storage. The key benefits are user governance, high-control combined with self-service capabilities, auto-scaling, auto-healing, and auto-provisioning. This reduces the overall maintenance cost of running these solutions and allows an organization's DevOps team to maintain these applications themselves.

Added Value

Agile platforms will offer out-of-the box tools like automated testing and application monitoring. Automating functional tests for the user stories you are developing can reduce overall cost of testing to less than 10% of your project budget. More importantly, governance tools, like application and portfolio quality monitoring, give instant insight into the quality of your portfolio, reducing lifecycle costs as well as the effort associated with app/portfolio rationalization.

Business Alignment

When you run projects, you want your makers and business leaders talking directly to each other to eliminate the number one reason why projects are failing: lack of user input and unclear or changing requirements. With the right low-code development platform, project team members can connect instantly and react to changes and updates such as new requirements, software revisions, changing vision, as well as test results and feedback from end-users.

Visual models and drag-and-drop WYSIWYG editors lower the barrier to entry for business users to participate in the development process, enabling cross-functional teams to close the feedback loop[8] by working collaboratively and iteratively while accelerating the development process towards results that actually deliver business value.

Extended Developer Resources

Low-code development environments provide visual, drag-and-drop tools for UI, data, logic and navigation to support a spectrum of users who are necessary for delivering value in today's software-driven world. This includes embracing all developers, including professional developers, line-of-business developers, citizen developers, and business analysts.

The right low-code platform will enable each of these roles to deliver work through an integrated set of tools across the entire product lifecycle. The result is building solutions that are specifically tailored to the organization's business process without the need for significant retraining or hiring.

<CHAPTER 06>

Low Code is the Real Deal

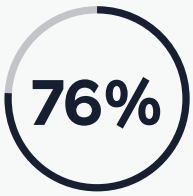
Achieving the Better Build

Before low-code development, the options were to buy and win in time-to-market but lose in agility or build and win with agility and lose in time-to-market. Now, low code enables organizations to differentiate with the combination of agility and fast time-to-market while still managing costs.

A survey[9] of 370 customers conducted by FileMaker found that 91% of the respondents building custom applications with low code reported an increase in productivity, with more than half reporting productivity gains of over 40%. Overall, the survey finds that 76% reported gaining a return on investment (ROI) on those projects, and 77% saw a reduction in manual data entry.

Increase in Overall Productivity

Reduction in Manual Data Entry



Gain a Return on Investment

Let's see what that looks like in the real world.

Imagine a mid-sized manufacturing company has a large SAP install base, but the salespeople use Salesforce. The field workforce for the company deliver and service the product that they are manufacturing. Because connectivity on customer sites is patchy at best they still use paper forms out in the field. Then they transpose the data from the form into the system at the office.

Now think of that same field worker onsite with a customer. The customer wants to order new parts, but the field worker doesn't have access to SAP in the field. So the field worker has to either refer the customer to their sales rep or write down the order request and process it back at the office. A sale fully processed by a field worker might also bypass Salesforce entirely, so sales activities and metrics will be inaccurate.

This is a typical situation for most organizations involved in field service. The mix of systems and manual and digital processes is inefficient and prone to error.

Think about what is now possible if they invest in the right combination of low code, serverless, PaaS, and SaaS.

The manufacturing company can:

- **Build a simple native mobile application, that works offline, with a low-code platform for data entry to optimize the efficiency of field workers.**
- **Integrate SAP and their inventory data into the mobile app, enabling field workers to be able to process sales remotely.**
- **Use serverless processes to sync bulk order changes from SAP with the app.**
- **Use low code to extend Salesforce to interact with SAP and the mobile app so all sales are tracked accurately.**

Now the manufacturing company has a rich and robust ecosystem leveraging serverless, SaaS, PaaS and legacy systems that are all tied together to drive additional revenue. The company is saving costs due to less rework to account for manual errors and field workers aren't wasting time on data entry. Instead they can now optimize their time selling and maintaining products.

Not only did they fix existing problems, but they also made it possible to leverage new technologies that would significantly improve customer experience and profit (e.g. Augmented Reality, AI/ML algorithms, etc.).

The best part is that the above example is not just theoretical. Real businesses have followed this exact progression and are now reaping the benefits. Read more about how Saga Healthcare[10], BAM Infra[11], and PostNL[12] shifted their paradigm with low code.

Saga Healthcare

- Launched a functioning app in 6 months
- Increased scheduling efficiency by 10%
- Gained 2.5 years in market

BAM Infra

- Created end-to-end smart meter installation app
- Integrated new apps with S/4HANA database
- Built 3 apps in 6 months

PostNL

- Built a core supply chain system
- Processes 10 million transactions per day
- Have 4.5 million users on package-tracking app

This is what is now possible, and the results speak for themselves.

<CHAPTER 07>

Finding a New Balance

Think Agile, Get Flexible, Evolve Always

Build is now the answer, but not in the way you think...

Low code and the synergy it has with new innovative technologies, like serverless computing and SaaS and PaaS solutions, gives businesses the best of both worlds. Organizations can use these new technologies to construct agile IT ecosystems with flexible and secure customizations that connect business-critical core software with the thrilling potential of AI, ML, and predictive analytics.

Through a thoughtful balance of build and buy your organization can continually evolve, upgrade, and innovate. With low code you can connect disparate systems, gain a shaper competitive edge through new tech, and adapt to a landscape of constant disruption. All of this without adding to your legacy and technical debt burdens.

Your organization must challenge conventional wisdom and traditional choices to thrive and compete.

...Are you ready?

Building the Paradigm

Curious about the business impact of built software?

Find out what companies like Zurich Insurance and SUEZ accomplished with **rapid app development**—faster launch, less money, and better results are just the beginning.

The Immediate ROI of Build

One great example of an organization achieving a huge ROI by building with a low-code platform is NC State University[13]. They needed a learning management system, and instead of buying a COTS solution like Blackboard, they built an app using the Mendix low-code platform.

The university doesn't receive federal dollars to support the management of a non-credit system and couldn't afford the time or the money it would cost to develop a custom solution in-house, so it released an RFP. The response was staggering. Estimates for an application ranged from $3-10 million over a five-year period.

"We absolutely could not afford that, so we charged our student interns with the task of researching other options. That's when we started to become aware of low-code platforms as an alternative way to rapidly develop and deliver code."

Gwen Hazlehurst, *Assistant Vice Chancellor, Enterprise Application Services at NC State*

Using Mendix, the newly-hired developers built a minimum viable product of a non-credit enrollment system, dubbed REPORTER, in just over two months, with the full launch occurring in just five months. The total cost for REPORTER is estimated in the low six figures, versus $3-10 million and five years quoted by COTS vendors. What's more, the NC State team has greatly expanded REPORTER's functionality, which would've been incredibly costly, time-consuming, or impossible with a COTS product.

> **"Thirty or so departments use REPORTER, and they've all realized FTE savings. They've all been able to take that burden of worrying about what they're going to do next and how they'll manage an internal application, and reallocate those resources."**

Jack Foster, IT Director at NC State

COTS Vendor Estimate

Cost: $3–$10 million

Time to Market: 5 years

Mendix low-code platform

Cost: Low 6 figures

Time to Market: 6 months

The ROI of low-code development wasn't limited to the REPORTER build for NC State. One of their developers, Jordan Boyle[14], started as a student intern on the IT team. He was a Finance student and after graduation was hired as a full-time developer. Despite lacking a traditional programming background, he is among their most productive developers. He built another app in Mendix that addressed the problem of technical equipment usage at the University. That app generates $1 million in annual revenue.

[1] "The Chaos Report," The Standish Group, 1995.

[2] "2009 Business Analysis Benchmark Study: The Path to Success," IAG Consulting, 2009.

[3] Goodman, Danielle. "Jordan Boyle: From No Programming Background to Sole Developer of NCSU's Lab Management App." Mendix.com/Blog. October 4, 2018.

[4] Shelton, Cindy. "Reengineering with Commercial Off-the-shelf-software," Defense AT&L. Business Process, September – October 2010.

[5] "2009 Business Analysis Benchmark Study: The Path to Success," IAG Consulting, 2009.

[6] Moore, Susan. "Gartner Says Demand for Enterprise Mobile Apps Will Outstrip Available Development Capacity Five to One." Gartner Newsroom. June 16, 2015.

[7] Brogan, Neil. "Are Your Struggling to Hire a Team of Software Developers?" iTech. August 1, 2017.

[8] Koelewijn, Andrej. "Flexible Integration Options to Support Next Generation Low-Code Applications." Mendix.com/Blog. July 19, 2018.

[9] Goodman, Danielle. "Agile Process: Why You Need Fedback Loops Both During and After Sprints." Mendix.com/Blog. April 17, 2018.

[10] "Low-Code Platforms Shift Application Buy Versus Build Debate." ITBusinessEdge.com.

[11] "A Healthy Outlook: Saga Healthcare Disrupts Established Homecare Market with a Low-code Platform." Mendix.com/Customer-Stories.

[12] "The Distrupted Becomes the Disruptor: Multi-billion-euro Construction Company Transforms Itself into a Digital Native." Mendix.com/Customer-Stories.

[13] Bevans, Dave. "National Parcel and Logistics Companies Deliver Digital Transformation with Low-code." Mendix.com/Blog. June 11, 2019.

[14] "An Engine of Economic Growth: NC State Uses Low-code Development to Deliver 500,000 Non-credit Course Registrations per Year." Mendix.com/Customer-Stories.

mx mendix
Go make it™

About Mendix

Mendix is the fastest and easiest platform to build and continuously improve mobile and web applications at scale. Recognized as a market leader by leading analysts, we help our customers digitally transform their organizations and industries by building, managing, and improving apps at unprecedented speed and scale. More than 4,000 forward-thinking enterprises, including Kuehne + Nagel, Philips and North Carolina State University use our platform to build business applications to delight their customers and improve operational efficiency.

We made this for you with care (and lots of coffee).

Lead Writer
Danielle Goodman

Lead Designer
Sydney Moore

Editor
Taylor Bornstein