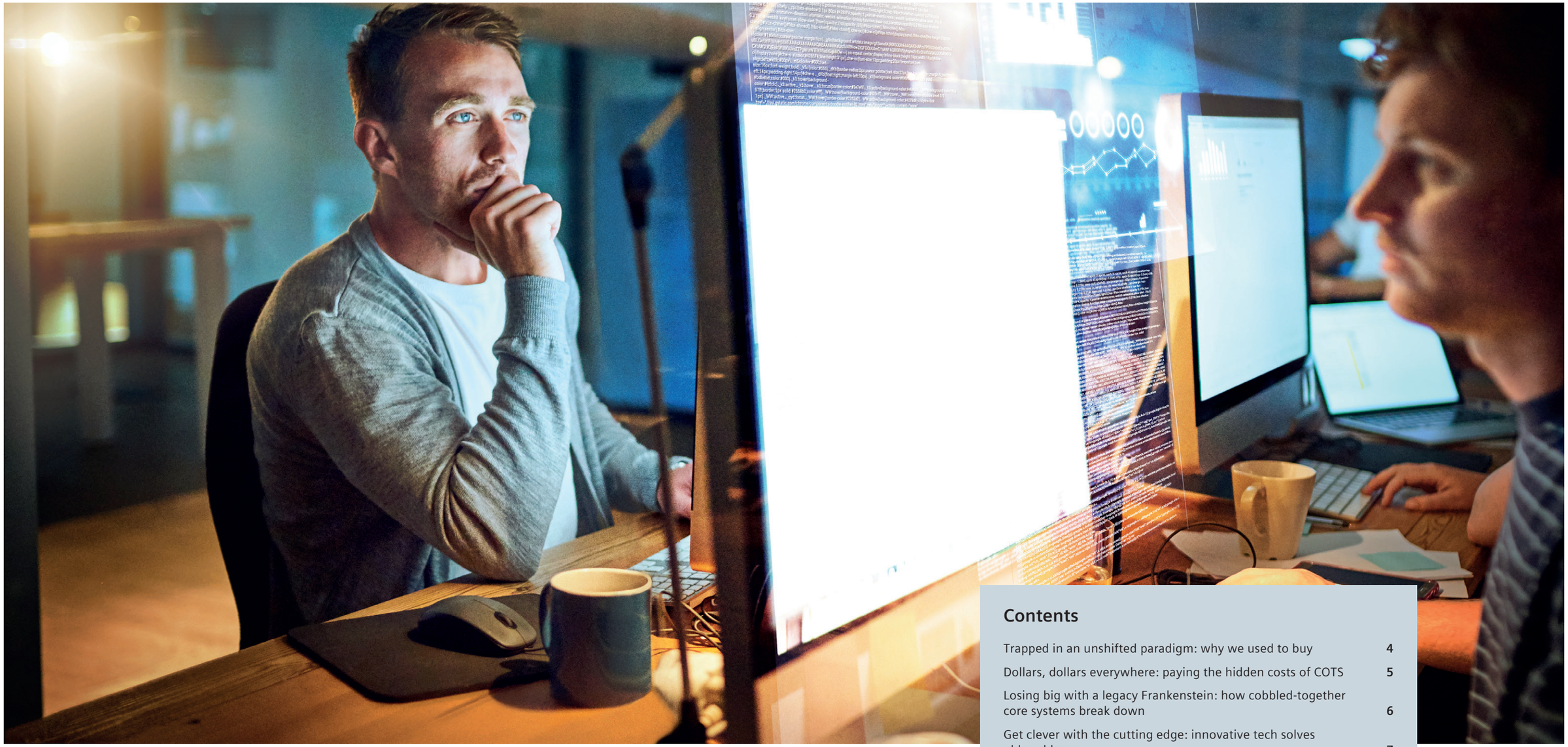

Siemens Digital Industries Software

# Picking a path: build versus buy

## Use low-code development to gain a competitive edge by adapting to constant disruption

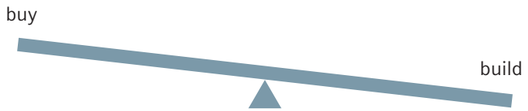
siemens.com/software



## Contents

# 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 costs (such as operational and maintenance, developer resources and other expenses required upfront), implementation concerns and lengthy time-to-market.

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

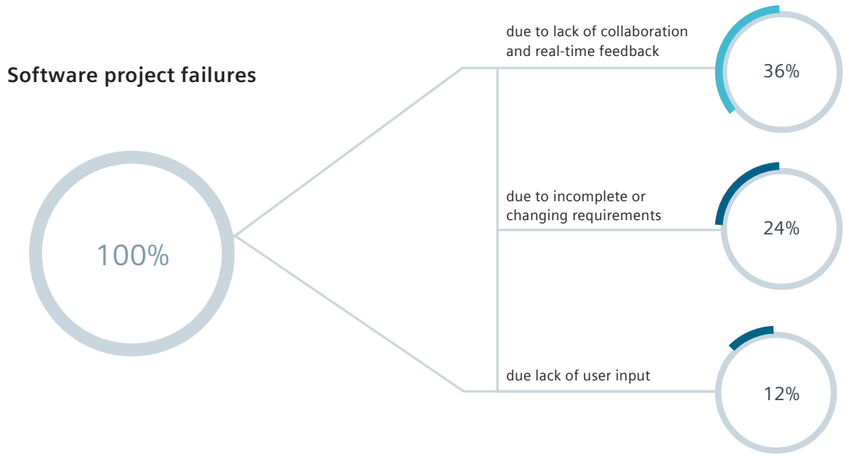
Follow-up research by IAG Consulting[2] in 2009 found that 68 percent of IT projects were failures from the start – doomed from the beginning by poor requirements. Fifty percent of the unsuccessful projects were considered runaways, all of them flagged as lost hopes due to at least two of the following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

- The project took nearly twice the estimated time to complete
- Costs exceeded more than 1.6x budget
- Less than 70 percent 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.**

**Software project failures**





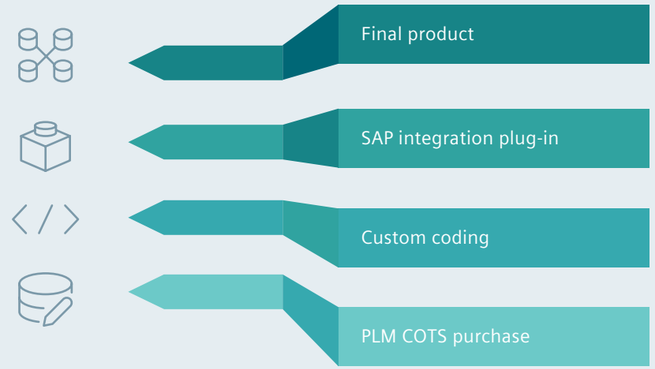

# Dollars, dollars everywhere: paying the hidden costs of COTS

In the last decade, the software market has undergone a transformation with significant consolidation. Long-standing players have disappeared. Now everyone has the same customer relationship management (CRM) system, enterprise resource planning (ERP) tool, and other core resources, such as SAP, Teamcenter® software and Salesforce. Organizations can no longer create a durable competitive advantage based on their core system choices. This is fundamentally different than it was just five years ago and is forcing businesses to change their thinking as it pertains to building versus buying.

The fantasy of COTS is that with a few plug-ins and bit of coding those solutions will neatly dovetail into your core systems. The reality is this – COTS software requires extensive customization, workarounds and creative rigging to achieve functional integration with an organization's information technology (IT) infrastructure. Implementing a COTS software solution always ends up with people "Frankensteining" their way to full functionality.

For example, say you buy a product lifecycle management (PLM) system to improve your IT service management, but 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 the implementation sets in you see the glaring need for software that sits between your core systems. The new PLM system must integrate with SAP to be able to continue to manage an existing process, which requires more customization.

**IT service management system**




However, the problem is the purchased solution isn't structured to support complex builds.

**Generally, purchased solutions get you 60 percent of the way there, but you have to build to get the other 40 percent.** 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 savings for 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 lifecycle of the products, we cannot allow any modification."

*Cindy Shelton*
*Business Process Reengineering with Commercial Off-the Shelf Software*[3]

# Losing big with a legacy Frankenstein: how cobbled-together core systems break down

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

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 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.

Todays' business can't afford to lose time or money buying and implementing the same solutions their competitors are using.




# Get clever with the cutting edge: innovative tech solves old problems

Companies that are investing in cloud, agile method of development and DevOps are on a glide path to low-code development because the convergence of all these technologies and methodologies has made build a favorable option.

Similarly, with developments in serverless, platform-as-a-service (PaaS), and software-as-a-service (SaaS) businesses can enable a competitive advantage by combining these technologies to optimize processes while mitigating all the concerns that were once associated with building custom software.

1. **Core systems are open and have rich interfaces** so you can build software that lives in between them and build fit-for-purpose solutions.
2. **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 technologies have also democratized the role of the developer. According to IAG Consulting research[4] from 2009, 70 percent of companies stated 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.

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 (prodev) talent – 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 eight to 12 weeks or longer hiring a specialized team of developers.

The solution to the paucity of professional developers isn't an exorbitant IT payroll. The answer is to empower citizen developers (those without a technical/programming background) throughout your organization to contribute to the building process using the new, more accessible tools available.

**Building is no longer the exclusive purview of the professional developer.**

# 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 and cons in the conversation around enterprise software acquisition have changed.

### Innovative integration

Purchased software doesn't enable easy integration with new technologies like cognitive services, machine learning algorithms or augmented reality (AR). The right cloud and low-code platforms support integration[7] with these innovative technologies and services using 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. 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 percent 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 fail: the lack of 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 deliver business value.

### Extended developer resources

Low-code development environments provide visual, drag-and-drop tools for the user interface (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.

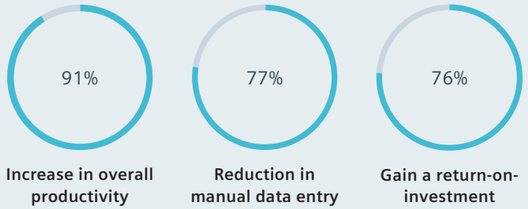
The right low-code platform will enable each of these roles to deliver work through an integrated set of tools across the entire product lifecycle. The result is building solutions that are specifically tailored to the organization's business process without the need for significant retraining or hiring.



# 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 percent of the respondents building custom applications with low code reported an increase in productivity, with more than half reporting productivity gains of over 40 percent. Overall, the survey found that 76 percent reported gaining a return-on-investment (ROI) on those projects, and 77 percent saw a reduction in manual data entry.




### 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 they are manufacturing. Because connectivity on customer sites is patchy at best they still use paper forms out in the field. Then they copy 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 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 (for example, augmented reality (AR), artificial intelligence (AI)/ machine learning (ML) algorithms, etc.).

The best part is 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 percent
- Gained 2.5 years in market

**BAM Infra**
- Created end-to-end smart meter installation app
- Integrated new apps with S/4HANA database
- Built three apps in six months

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

# 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.

With 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 with new technology and adapt to a landscape of constant disruption. You can achieve 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.**

# The Immediate ROI of build: use case

One great example of an organization achieving a huge ROI by building with a low-code platform is North Carolina State University (NC State).[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 a request for proposal (RFP). The response was staggering. Estimates for an application ranged from $3 to 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," says Gwen Hazlehurst, assistant vice chancellor, enterprise application services, North Carolina State University. "That's when we started to become aware of low-code platforms as an alternative way to rapidly deliver code."

"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 deliver code."

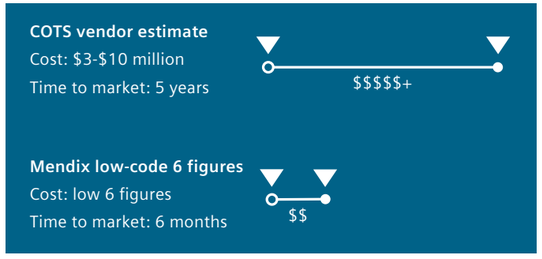
*Gwen Hazlehurst*
*Assistant Vice Chancellor, Enterprise Application Services*
*North Carolina State University*

Using Mendix, which is part of Xcelerator, the comprehensive and integrated portfolio of software and services from Siemens Digital Industries Software, 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 the $3-to-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 costly, time-consuming or impossible with a COTS product.

"Thirty or so departments use REPORTER, and they've all realized FTE savings," says Jack Foster, information technology director, North Carolina State University. "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."

"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*
*Information Technology Director*
*North Carolina State University*




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.



# References


1. "The Chaos Report," *The Standish Group*, 1995.
2. "2009 Business Analysis Benchmark Study: The Path to Success," *IAG Consulting*, 2009.
3. Shelton, Cindy. "Reengineering with Commercial Off-the-shelf-software," *Defense AT&L. Business Process*, September-October 2010.
4. "2009 Business Analysis Benchmark Study: The Path to Success," *IAG Consulting*, 2009.
5. Moore, Susan. "Gartner Says Demand for Enterprise Mobile Apps Will Outstrip Available Development Capacity Five to One," *Gartner Newsroom*, June 16, 2015.
6. Brogan, Neil. "Are Your Struggling to Hire a Team of Software Developers?" *iTech*, August 1, 2017.
7. Koelewijn, Andrej. "Flexible Integration Options to Support Next Generation Low-Code Applications," *Mendix.com/Blog*. July 19, 2018.
8. Goodman, Danielle. "Agile Process: Why You Need Feedback Loops Both During and After Sprints." *Mendix.com/Blog*. April 17, 2018.
9. "Low-Code Platforms Shift Application Buy Versus Build Debate," *ITBusinessEdge.com*.
10. "A Healthy Outlook: Saga Healthcare Disrupts Established Homecare Market with a Low-code Platform," *Mendix.com/Customer-Stories*.
11. "The Distrupted Becomes the Disruptor: Multi-billion-euro Construction Company Transforms Itself into a Digital Native," *Mendix.com/Customer-Stories*.
12. Bevans, Dave. "National Parcel and Logistics Companies Deliver Digital Transformation with Low-code," *Mendix.com/Blog*. June 11, 2019.
13. "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*.
14. Goodman, Danielle. "Jordan Boyle: From No Programming Background to Sole Developer of NCSU's Lab Management App," *Mendix.com/Blog*. October 4, 2018.


**About Siemens Digital Industries Software**
Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Xcelerator, the comprehensive and integrated portfolio of software and services from Siemens Digital Industries Software, helps companies of all sizes create and leverage a comprehensive digital twin that provides organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.


| | |
|---|---|
| Headquarters: | +1 972 987 3000 |
| Americas: | +1 314 264 8499 |
| Europe: | +44 (0) 1276 413200 |
| Asia-Pacific: | +852 2230 3333 |



81830-C2 7/20 H