



The Higher Ed Guide to Automation and Workflow Tools

KUALI BUILD



Contents

03 Introduction
The Future of Campus Work Is Now

04 Chapter 1
What is Process Automation (Really?)

05 Chapter 2
The Hidden Cost of Paper, PDFs,
and Email

06 Chapter 3
No-Code vs. Low-Code—Why It
Matters More Than You Think

08 Chapter 4
The Central IT Struggle
(And How We Help)

09 Chapter 5
Integrations Without the Headaches

10 Chapter 6
Built-In Trust—Security and
Compliance, Demystified

11 Chapter 7
Start Small. Think Big.



INTRODUCTION

The Future of Campus Work Is Now

Every day on campus, there's a quiet avalanche of admin work: forms passed around in emails, signatures tracked down by phone, spreadsheets duplicated and re-saved with names like "FINAL_final_revised_v3." It's tedious, fragile, and—let's be honest—a little absurd in 2025.

Meanwhile, higher ed is changing fast. Budgets are tightening. Programs are under pressure. Positions aren't being backfilled. And the folks still standing? They're being asked to hold it all together with duct tape and goodwill.

We see you. And we built this guide with you in mind.

"Digital transformation" might sound like something cooked up in a conference room—abstract, expensive, and faintly terrifying. But here's the thing: clinging to outdated tools is already costing more than you think. In time. In morale. In missed opportunities to do the work that actually matters.

Change doesn't have to start with a big IT initiative or a massive overhaul. Sometimes, it starts with fixing that one approval process everyone dreads.

This guide is for HR leaders, registrars, deans' assistants, and every unsung hero who's keeping the gears turning. It's packed with practical ideas, real-world examples, and a few moments of "Okay, that's exactly what I'm dealing with."

As one of our users from UC Riverside put it: "We were just trying to solve a couple of nagging problems. What we ended up with was a better way to work across the board."



Chapter 1

What is Process Automation (Really?)

Let's clear something up: process automation is not an IT project in disguise. It's not about replacing people or rolling out some giant system you'll spend months training on. It's about working smarter.

Every department on campus—from HR to academic affairs—relies on processes. These are the routine steps, handoffs, and approvals that keep things running. Faculty credential reviews. Student petition approvals. Curriculum changes. New employee onboarding. You know the ones.

And most of them? Still handled with a clumsy mix of PDFs, spreadsheets, inboxes, and “Did you get this yet?” hallway conversations.

That's where process automation steps in. Instead of hunting for signatures or guessing who's next in the chain, your system handles it. You define the process, and it follows through. Automatically.

A Real Example: Student Petitions

Imagine a student wants to drop a course. They print out a form, scribble their name, and carry it—by hand—to three different offices. Someone's out that day. The form sits. Maybe it makes it back. Maybe it doesn't.

Now picture this instead: the student fills out a quick form online. It goes straight to their advisor. When approved, it automatically moves to the department chair, then the registrar. Everyone gets notified. Everyone sees where it's at. Nobody wonders what happened to it.

Why This Matters

Campuses today are asked to do more with less—less staff, less time, less funding. Automation doesn't mean adding another system. It means removing the chaos. No more waiting on replies. No more copy-pasting. No more re-entering the same data across multiple systems. It's not about high-tech wizardry. It's about breathing room. About getting an hour back. About fixing one annoying



thing and suddenly having time for the next.

Most of us aren't dying to learn a new platform. That's why tools like Kualu Build are designed for real humans. If you can fill out a form, you can automate one. And here's the key takeaway: automation doesn't replace people. It replaces the burnout, the bottlenecks, and the broken workflows keeping good people from doing their best work.

Chapter 2

The Hidden Cost of Paper, PDFs, and Email



Next Up: low-code vs. no-code—what's the difference, and why does it matter more than you think.

Let's talk about the stuff nobody budgets for: the endless admin work that flies under the radar. The shared drives. The forgotten attachments. The "URGENT" subject lines on emails sent into the void.

On the surface, it seems harmless. Maybe even efficient, in a scrappy way. But those familiar workarounds? They're stealing hours, delaying decisions, and fraying nerves.

Every time someone manually retypes a name, prints a form, or sends a reminder email, it's time lost. Time that could have gone to a student. Or a colleague. Or a problem that actually matters.

"We were spending so much time chasing people down, we almost forgot what the actual goal of the process was." — Radford University

The cost isn't just in morale. It's in real dollars, too.

You're paying for things that don't show up as line items, but add up fast:

- Paper, toner, and storage for physical forms
- Staff hours spent onboarding people into patchwork systems
- Rework caused by lost forms or outdated versions
- IT time spent patching together tools they never approved
- You're paying when a form gets lost and someone misses a deadline.
- When a broken macro brings everything to a halt. When people get so frustrated, they invent their own version of the process. Again.

And the worst part? It's invisible on the budget sheet. Because there's no line item for wasted potential.

Now imagine a different scene.

A form goes out digitally. It's filled out once. No printing. No chasing. Everyone who needs to sign it gets pinged automatically. You can check the status anytime—like tracking a package. And when it's done? The data goes where it needs to go. No re-entry. No "last known version." Just done.

That's not hypothetical. That's how dozens of institutions are using Quali Build today. And they're doing it without hiring developers or launching six-month projects. You don't need a transformation plan. You need one small win. Let's start with that.



Chapter 3

No-Code vs. Low-Code—Why It Matters More Than You Think

If you've ever been promised an "easy-to-use" form builder—only to find yourself waiting two weeks for IT to troubleshoot a dropdown menu—you already know the difference between no-code and low-code. On the surface, these terms sound like marketing fluff. But for people actually doing the work? They mean everything.

What Low-Code Really Means

Low-code tools often promise flexibility and power. And they deliver—if you have someone who can write JavaScript, understand API calls, and manage technical debt. But here's what usually happens instead:

- A department admin builds a form, gets stuck halfway, and sends it to IT.
- IT, already overextended, tries to clean it up.
- Bugs creep in. Data doesn't map cleanly. Testing takes longer than expected.
- Everyone's frustrated. Deadlines slip. Trust erodes.

As one IT manager told us: "Low-code means we end up owning someone else's half-baked idea."

It's not that these tools are bad. It's that they shift technical burden to a team that's already underwater.

What No-Code Actually Feels Like

Now let's talk about no-code. Real no-code. The kind where you don't need to understand how an integration works, just that it does.

You drag and drop. You set up your own logic. You test it, tweak it, and launch—all without a ticket to IT.

With Kuali Build, we've seen HR teams launch onboarding workflows on their own. Registrar's offices build curriculum change forms. Student affairs staff create travel requests.

And they maintain them—without roping in IT every time someone retires or a policy shifts.

"We needed something we could own. Something we wouldn't break every time we changed a rule. Kuali Build gave us that." — Director of Enterprise Applications, Cal State Bakersfield

No-code means fewer shadow systems, fewer fire drills, fewer support tickets. It gives IT time back—so they can focus on infrastructure, security, and the big stuff that only they can do.

In short: low-code promises freedom, but often delivers dependency. No-code, when done right, delivers autonomy—for both admins and IT.

Next Up: how to keep that autonomy intact with integrations that actually work, without drowning in middleware.

A DAY IN THE LIFE OF ALEX

Let's say Alex is in charge of onboarding new student workers.

WITH LOW-CODE PLATFORM	WITH KUALI BUILD
Alex gets a starter template, makes a few changes, but hits a snag with conditional logic. IT jumps in, fixes the issue—but then the form breaks again when someone edits a field down the line. Three weeks pass before it's live.	Alex drags in a date field, sets up logic with a dropdown, adds an automated notification, and tests it in under an hour. It works. She shares it with her team that afternoon.

*Same Alex. Same goal.
Different outcome.*

KUALI
BUILD





Chapter 4

The Central IT Struggle (And How We Help)

Let's be honest: central IT has been carrying more than its fair share for a long time.

It's not just keeping systems running or managing security threats—it's also answering support tickets for tools you didn't buy, cleaning up broken integrations you didn't configure, and chasing down data inconsistencies caused by... well, everyone else's best intentions.

The decentralization of tech across campus was supposed to be empowering. But in reality? It's often meant that IT is stuck in the middle—responsible for the stability of systems no one asked you to vet.

"It's not that we don't want people to build their own tools. We just don't want to spend weeks cleaning them up afterward." — Director of Enterprise Applications, UC System

Kuali Build gives you a way out of that loop.

It's a no-code platform that allows departments to build and manage their own digital processes—within an ecosystem that IT controls. That means governance without micromanagement. Autonomy without chaos.

Instead of reacting to another "Hey, can you make this work?" email, you set the guardrails up front. Everyone builds inside a vetted, secure, integrated system. You stay ahead of the shadow IT problem—not buried under it.

And here's the twist: this isn't just about protecting

your time. It's also about elevating your role. Because when you're not buried in reactive support work, you finally have the bandwidth to:

- Lead strategic data initiatives
- Modernize infrastructure
- Support innovation at scale

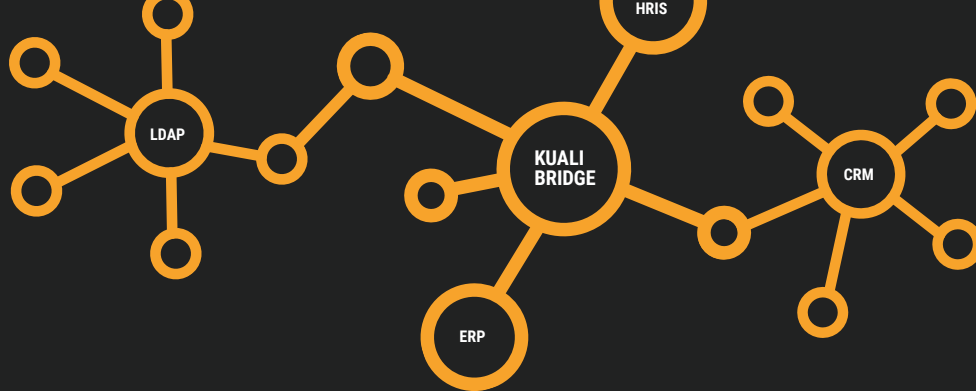
Kuali Build doesn't just lighten your load. It expands your impact.

And here's where it gets interesting: Kuali Build doesn't just compete with low-code solutions—it does what they do. Full stop. Complex branching logic? Check. Role-based permissions? Built in. REST APIs, real-time integrations, and robust data routing? All there. You can even plug directly into tools like ChatGPT and Claude to generate dynamic instructions or process content on the fly. Just drop in your key and go. And for the real power users? Kuali supports GraphQL queries and offers full support for advanced integration scenarios.

But if the data you need is fragmented, inconsistent, or lives across multiple systems (and let's face it, it usually does), that's where Kuali Bridge comes in.

Bridge is our next-generation integration tool—a visual, node-based interface that lets you transform, enrich, and orchestrate data without writing middleware. It's like if an API gateway and a translation engine had a baby, and that baby spoke fluent higher ed. We'll dig into how it works—and why it matters—in the next chapter.





Chapter 5

Integrations Without the Headaches

This is the problem Kuali Bridge was built to solve.

Bridge is a behind-the-scenes translator. It helps your systems talk to each other—even when they speak different languages or only share half the information you need. It's like a universal adapter for your campus data.

If you're not technical? Just know this: with Bridge, your forms and workflows can finally connect with the tools you already use—securely, reliably, and without endless back-and-forth.

If you are technical? You'll appreciate this:

- Drag-and-drop, node-based visual builder
- Pull data from REST or GraphQL endpoints
- Transform and enrich responses in-flight
- Push structured results into your SIS, HRIS, CRM, or ERP

So, whether you're trying to pre-populate a form from Banner or push data back into Colleague, Bridge lets you do it—all without writing or maintaining custom middleware.

"With Kuali Bridge, we finally had a way to integrate data from across campus without burning three weeks of dev time." — Lead Systems Integrator, Regional University

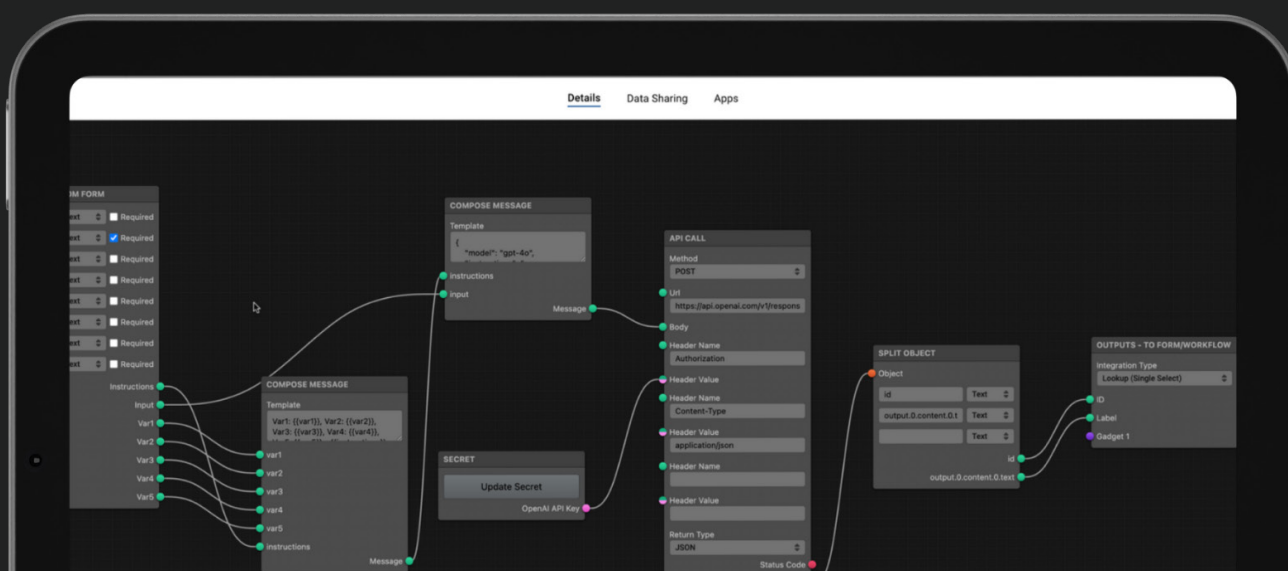
This isn't just nice-to-have. It's how IT stops being the bottleneck, and departments stop working in silos. It's how digital transformation actually gets done.

Because Bridge was built as part of the Kuali Platform, it speaks the same language. No translators, no one-off connectors. Everything works together. It's purpose-built for higher ed, which means it's built for you.

So yes—your digital forms can talk to your systems. And they can do it:

- Automatically
- Securely
- Without surprises

Next up: let's talk about how Kuali Build keeps all of this secure and compliant—without creating extra work for IT, legal, or procurement.



Chapter 6

Built-In Trust—Security and Compliance, Demystified

Let’s face it: the moment someone on your team says, “We want to try this new tool,” the very next question is usually, “Has IT approved it yet?”

And that’s fair. Because higher ed has a responsibility—to students, to staff, and to your institution’s reputation—to take data privacy and security seriously. Especially when it comes to workflows that deal with personal, financial, or academic information.

But that responsibility often turns into roadblocks.

Security questionnaires. Vendor vetting. Procurement slowdowns.

That’s why Kuali Build shows up prepared. We’ve already cleared the hurdles that usually stall projects before they start:

We’ve already cleared the hurdles that usually stall projects before they start.

SOC 2 TYPE 2 COMPLIANT	HECVAT COMPLETED	ENTERPRISE-GRADE CLOUD INFRASTRUCTURE
This means we’ve been independently audited to ensure our systems are secure, available, and handle your data responsibly. Translation? Your data is safe.	The Higher Education Community Vendor Assessment Tool (HECVAT) is the gold standard for vetting software vendors in higher ed. We’ve already done the paperwork, so your procurement and IT teams don’t have to start from scratch.	Hosted on secure platforms with built-in redundancy, encryption, and uptime that meets the demands of mission-critical campus operations.

"As soon as I saw they'd already completed a full HECVAT and had SOC 2 in place, I knew we could move fast without skipping any of our security steps."
— CIO, Public Liberal Arts College

Here’s the bottom line: Kuali Build is ready for prime time. It’s not a startup experiment. It’s not “freemium” software that makes your security team sweat. It’s a platform designed from the ground up to support

higher education’s real-world requirements.

Whether you’re in IT, legal, or procurement, or you’re just trying to get a project approved, Kuali Build takes the risk off the table so you can focus on results.

Next up: Let’s take a look at real-world stories from universities that’ve already made the leap—and what happened next.

Chapter 7

Start Small. Think Big.



Here's the thing: no one is handing out million-dollar digital transformation grants. You're not sitting on a surprise budget. And you probably don't have a change management team just waiting in the wings. That's okay.

You don't need a big team or a giant initiative to get started with Kuali Build. You need a problem. A champion. And a few people who are tired of saying, "There has to be a better way."

Start with a form that's driving everyone a little bit nuts—an outdated process, an approval loop that never ends, or a PDF so old no one remembers who built it.

Map out the current steps. Talk to the people involved. Rebuild it digitally. Pilot it with your team. Then expand from there.

And when you're ready to think about scaling, we've got you covered there too. Our companion guide, *Strategic Rollout Models for Kuali Build*, breaks down eight proven strategies drawn from real institutions:

Open Enrollment Implementation	Invite everyone, and let usage grow organically.
Iterative Departmental Expansion	Start with one unit, then expand strategically.
Distributed Governance Model	Empower departments with autonomy under IT guidance.
Executive Mandate Rollout	Drive change top-down with clear leadership backing.
Power User Adoption Strategy	Identify champions and let them lead grassroots adoption.
IT-Managed Service Model	Keep control centralized, with high standards and consistency.
Hybrid Rollout With Delegated Access	Start with IT, then shift ownership as users gain confidence.
Autonomous Departmental Rollout	Let individual departments own and manage their own use of Build.

Whether your institution is highly centralized, fiercely decentralized, or somewhere in between, there's a model that fits your style—and your bandwidth. You don't have to boil the ocean. You just have to start small, start smart, and keep going.. You just have to start. And when you do? We'll be right here with you.



LEARN MORE FROM **KUALI** USERS

Want to learn more from our coast-to-coast university success stories? Check out case studies and resources here.



kuali