ERP SCORECARD MASTERCLASS
Choose the Right Software with Confidence

HOST (BRAYLEE)
All right, good afternoon. Thank you, everybody, for joining us for our webinar: *ERP Scorecard Masterclass – Choose the Right Software with Confidence*.

Before we get started, just a quick housekeeping note: all participants will be muted. But if you have any questions, feel free to drop them into the Q&A at any time—before, during, or after the presentation. And I stress *during*—please don't be shy!

Our presenter today is Steve Weber, CEO and co-founder of PaperTrl.
So Steve, I'll hand it off to you.

PRESENTER (STEVE)
Thanks, Braylee. Can you hear me okay?

HOST (BRAYLEE)
Yep!

PRESENTER (STEVE)
All right, perfect. I'm just going to share my screen here...

[Pauses to share screen]

Okay, just a little background on me—so as Braylee mentioned, I'm the co-founder and CEO of PaperTrl.

PaperTrl is an AP automation company that integrates with a wide variety of ERP systems. Much of what we do hinges on providing connectivity to those systems so we can collaborate not only with accounting teams, but with departments across an organization.

Over the years, we've had the opportunity to work with many different ERP systems. Customers often bring us requests for systems they want us to connect with—and we've learned firsthand that some are far better than others when it comes to integration.

That experience really inspired us to create this buyer's guide—because at the end of the day, our goal is to help as many companies as possible automate their AP processes. And

that's simply easier when businesses are using modern, open, and integration-friendly software.

What we've also discovered is that it doesn't just make life easier for us as a vendor—it actually benefits the companies themselves. Many organizations try to roll out new solutions or improve internal processes, only to hit a wall because they can't integrate with their accounting system.

So, over the last seven or eight years, I've done quite a bit of research in this space. I'm often the one reaching out to ERP vendors to explore integration options. And what you'll see today is based on both my personal experience and that of our team working with the systems we'll cover.

I should mention—this isn't an exhaustive list of every ERP we've worked with. Rather, it's a snapshot of the platforms that come up most frequently in our conversations with clients and partners.

That said, the evaluation criteria we'll walk through today are universally applicable, regardless of whether the ERP you're considering is on our list.

PRESENTER (STEVE)
So I'll tell you—my frustration with trying to help companies expand beyond the limits of their ERP system started years ago. But probably the most *poignant* example—one that really captures how limiting some systems can be—was with a family of automotive dealerships here in Central Ohio.

This story is what I like to call: *"The Revenue That Never Was."* It's about how an *industry-leading*—and I use that term intentionally, because half the industry used this ERP—how that system's *inflexibility* ended up blocking millions in potential sales for this company.

Here's the backstory:

This dealership group had more than a dozen locations throughout Central Ohio—and really, across the Midwest. They had tens of millions of dollars in auto parts inventory. They were one of the largest OEM part buyers in the U.S., buying from GM, Ford, Chrysler, Toyota—you name it. High-volume parts, fast-moving items, and rare, hard-to-find ones too.

If you needed a transmission for a '74 Chevy Vega?
You went to *this* dealership.

So what did they want to do?
With all that inventory, they had a clear vision: become the largest seller of trusted OEM

parts on eBay. And they brought our company in—this was back when I was at a company called Channel—to help make that happen.

To expand into the eBay marketplace, we needed **real-time integration.**
That was absolutely critical. Because all these dealerships were pulling from a *shared* inventory pool that was constantly in flux.

We couldn't risk listing something online that had just sold inside a dealership an hour earlier.

But here's what happened:
That real-time integration?
Total no-go.
The ERP system—again, an industry-leading platform—had *no way* to connect to their live inventory. The best we could do was export a daily snapshot of inventory levels. But we couldn't update it dynamically. We couldn't relieve inventory as online sales occurred. There was just… no path forward.

Even when we reached out to the vendor directly, they told us, "That's not something we support right now. It's something we're planning to include in a future release."

And so, in the end?
The dealership had to **abandon** their eBay expansion plans altogether.

Now, management *did* look at other ERP options—some with open APIs that could've made this work. But the alternatives? They were mostly newer players in the space. And while those platforms could've solved the problem, the idea of moving off an "industry-standard" system felt too risky.

The **conversion cost**—both in dollars and in disruption to their business—just seemed too high. Plus, moving to a new system would've meant hiring someone to manage the tech transition. That wasn't something this dealership group was equipped or willing to take on.

And then there was this constant *carrot and stick* dynamic from the ERP vendor:
"The feature is coming soon. Just wait for the next release."
And so, they waited.
And waited.
And that revenue opportunity slipped away.

So that's just one example—but honestly, stories like this are common. And they're a big part of why we put this scorecard together.

Today, what we're going to walk through is a framework—a scorecard—that helps companies evaluate ERP systems beyond the basics. Most ERP platforms can handle general accounting. That's not where the real differentiation lies.

What's harder to see upfront—and what becomes *way* more important over time—is how well your system supports growth, connectivity, and flexibility.

So we want to give you a bit of education on how to spot those differences early.

PRESENTER (STEVE)
Braylee, it looks like we have another attendee in the waiting room—can you let them in?

PRESENTER (STEVE)
The reason we felt the need to create this scorecard is because a lot of accounting systems—*even the laggards in the market*—they all use the same buzzwords. And they all tell the same story when you're evaluating them.

They're all *cloud-based*.
They're all *AI-powered*.
They're all *scalable*, *user-friendly*, with *business intelligence* baked in.

These are the terms you hear over and over again.

And frankly, when buyers are evaluating accounting systems, the core features have become so **commoditized** that everything starts to look the same. I mean, general ledger? Banking integration? Dashboards? Excel exports? That's all basically *pay-to-play* now in modern ERP packages.

So as a buyer, when you start comparing them at face value, they all look really similar. You bounce from website to website, and it's hard to tell which are the winners… and which are the ones you're going to regret investing in three to five years down the line—when you're ready to grow and suddenly realize you're stuck.

And maybe that's you. Maybe you're already *on* one of those legacy or laggard ERP systems and you're thinking:
"I know this software is bad. I know it's holding us back. But I don't exactly know *why*… or how we ended up here… or how to avoid this again."

That's what we're aiming to clarify today.

Now, when most companies go into the ERP selection process, they tend to focus on the hot-button items. And most buyer's guides reinforce that. They focus on things like:

- The user interface
- The aesthetic appeal
- Initial ease of use
- Upfront costs and the licensing model
- Vendor popularity—especially within a given industry

We see this all the time in industries like construction, automotive, real estate.
People ask around: "What's everyone else using?"

In real estate, for example, it's Yardy.
Everyone says, "You've got to be on Yardy. That's the standard."

And so decisions get made not necessarily based on *fit*, but just on what's *familiar*.
(For the record, we're *not* evaluating Yardy today.)

But here's the thing:
What's **seldom discussed**—and what becomes *critical* over time—is the **adaptability** of the software.
That's where this scorecard comes in.

Adaptability means: *How well can this system connect to the rest of your business ecosystem?*

Beyond that, companies also need to evaluate whether they're better served by:

- A **best-of-breed** solution, which is modular and can be tailored to your needs
  or
- An **industry-specific suite**, which tries to do it all in one package

Let me give you a few examples.

Industry-specific "all-in-one" platforms:

- *Deltek* – geared toward engineering firms, government contractors
- *Procore* – built specifically for the construction industry
- *CDK* – designed for automotive dealerships

These are large, purpose-built suites that attempt to cover every aspect of the business in one platform.

On the flip side, we have **best-of-breed** approaches, which let you assemble a tailored stack using powerful individual components.

For example:
Let's say you're in construction. You could use *Microsoft Business Central* as your ERP, then layer in Microsoft's add-on tools—like *Project Pro* for job costing, estimating, bidding, and managing onsite projects.

Need asset tracking?
Add an *asset management module*—and suddenly Business Central can do 90% of what a traditional construction ERP does, but with the flexibility to integrate and scale how you need.

Viewpoint, on the other hand, is an all-in-one ERP owned by Trimble, designed *exclusively* for construction. It bakes most contractor workflows into the core system. While it does support add-ons, the idea is: everything's already in the box.

PRESENTER (STEVE)
So when you think about all these factors, I like to visualize it in terms of **concentric circles**—layers of capabilities that define how well a software package serves your organization, both today and into the future.

Let me break it down into **three main capability layers**, plus a crucial fourth concept: **adaptability**.

---

1. Core Capabilities

This is the *innermost circle*. These are the foundational functions that **every** organization needs—regardless of industry. Whether you're a church, a car dealership, a manufacturer, or a city government, these are the must-haves.

Things like:

- Accounts Payable
- Accounts Receivable
- General Ledger
- Payroll

These are what I call **core capabilities**—they're non-negotiables. Every ERP should do these well, and they're not specific to your business model.

---

2. Productivity Capabilities

The next circle out—these are the tools that bridge **humans to work**. They allow your people to interact efficiently with the system, manage processes, and generate insights.

Examples include:

- Workflow engines (for routing tasks between users)
- Spreadsheet imports and exports (e.g., uploading or exporting a vendor list)
- Reporting tools
- Business intelligence dashboards
- Mobile apps

These capabilities help engage your team and make your processes more efficient. They're also where some systems start to differentiate.

---

3. Industry-Specific Capabilities

This is where software starts speaking *your* language.

Think of the features that Trimble builds into **Viewpoint** to make it a construction platform. These are the tools that help general contractors, electrical contractors, plumbing contractors—whoever—run their day-to-day.

Examples:

- Job costing
- Bid management
- Asset tracking
- Field crew scheduling

These are often **hard-coded** into the platform. You might be able to turn them on or off, or customize a field here or there, but this layer is part of the **DNA** of the system. It's what transforms a generic ERP into a vertical-specific solution.

---

4. Adaptability

Now here’s where things get interesting—and where long-term value is truly defined.

Adaptability is what determines whether your ERP can evolve with your business. It's the outermost circle—and, arguably, the most important.

This breaks into two areas:

---

a) Business Adaptability

This is your ability to **build differentiating processes** into the system. It’s about encoding your company’s *secret sauce*.

Maybe you send gift boxes to new customers. Maybe you have a special follow-up cadence. Maybe you have multi-step internal QA before an order is fulfilled.

Adaptability allows you to:

- Encode those custom workflows
- Ensure they scale as your team grows
- Make those processes *repeatable* and *reliable*

Without adaptability, your processes live in spreadsheets… or worse, in someone’s head.

---

b) Ecosystem Adaptability

This is about connecting to the **outside world**—the broader ecosystem that your business interacts with daily.

That could include:

- E-commerce platforms (like Shopify, eBay, Amazon)
- Supply chain partners
- Financial institutions
- Vendor portals
- Payment processors
- Banks (for issuing ACH, card, and check payments)

If your ERP can't connect to these systems, you're stuck. And that's what derails innovation.

In fact, it's usually these two adaptability layers—**business adaptability** and **ecosystem adaptability**—that ultimately determine whether your ERP has a **long-term future** in your business.

PRESENTER (STEVE)
Unfortunately, when companies go to purchase software—especially if they haven't had prior experience with ERP systems or don't fully understand the importance of adaptability—they tend to focus almost exclusively on **capability**.

Those are the hot-button issues.
That's what feels most urgent at the time.
And that's what the purchase decision is usually based on.

But then, after one year… two years… five years with the software, they start to realize something:

"All those core features—those capabilities—exist in a lot of different packages. But the **lack of adaptability**? That's what's killing us."

That's what leads to frustration.
That's what led the car dealership to abandon its plans to sell on eBay.
That's what we run into all the time when we start working with companies who want to automate their accounts payable—but we find we can't even **export bills** from their ERP or **send payment status** back in.

So that's why we created this **buyer's guide**.
And in putting it together, we didn't just want to offer opinions—we wanted to provide **objective**, experience-based criteria you can actually use to evaluate ERP systems.

Because let's be honest: there are plenty of buyer's guides out there already. But many of them:

- Focus solely on **capabilities**, not long-term value
- Are **behind paywalls** or part of expensive subscription services
- Accept **paid advertising** from the very vendors they're supposed to evaluate
- Or worse, allow vendors to **remove negative feedback**

That's not helpful to buyers. And it's not transparent.

And so, what we really want to provide here is a **subjective guide** based **on our own experience**. Everything I'm going to talk about today, just as legally suggested at the beginning, is based on emails, backups, communication for conversations we've had and our direct experience with these software packages.

So what did we do?

We took:

- All the ERP packages we've worked with
- A wide range of evaluation criteria
- And our knowledge of how these systems behave in the real world

And we asked ourselves:

- **What does "future-proof" really look like?**
- What are the market leaders doing to stay ahead?
- Who's following closely behind but doing well?
- Who's lagging—trying to catch up?
- And who has just… stagnated? Systems that haven't changed in five, ten, twenty, sometimes even thirty years?

That's the lens we're using today.
And again, every platform we'll talk about is one we've had direct experience working with.

Now in this session, we're **not** going to talk about evaluating the **capability** aspects of ERP systems. That's the stuff we talked about earlier—the inner circles: general ledger, reporting, workflows, etc.

There are tons of resources already available for that.
In fact, I've listed at least six right here on the slide.

But just a note of caution:
When you're looking at those resources, be aware of which sites **take advertising**, and which allow vendors to **scrub negative reviews**. Not all buyer guides are created with *your* best interest in mind.

So what *are* we going to focus on today?
We're going to talk about **adaptability**—specifically, how to evaluate ERP software for the **long-term**.

And that starts with architecture.

You might be wondering, *"Why does software architecture matter?"*
Maybe you don't know much about architecture—and that's okay. We're going to simplify it today.

But just like evaluating a house—you want to know whether the software has **good bones**.

Why?

Because good architecture:

- Makes it easier to **expand and scale**
- Simplifies integration with other tools and platforms
- Reduces both the **cost** and **time** it takes to implement new solutions
- Makes it easier to **change the system** without breaking something else
- And improves your experience with **add-ons and extensions** down the line

If the architecture is bad, even great add-ons can feel clunky or unstable.
If the architecture is solid, everything just works better.

PRESENTER (STEVE)
There's probably no better example of why architecture matters than comparing **QuickBooks Desktop** to **QuickBooks Online**.

Now, full disclosure—we are an Intuit partner.
We work with Intuit, and we're proud to be one of their partners.

But I'll tell you this—as an Intuit partner, I would **much rather** integrate with **QuickBooks Online** all day, every day, than with QuickBooks Desktop.

And here's why:

QuickBooks Desktop was originally built during the PC era—in the 1980s and 1990s. At the time, it was cutting-edge. And before cloud software was even a thing, I spent a lot of time building integrations to it.

But it came with *a lot* of issues.

For example:

- Integrations would randomly break
- They had to run on a **specific user's PC**, which meant if that person shut down their computer overnight, the integration would fail
- It was complex, fragile, and required manual oversight

Cloud architecture has changed all that. It's made integrations:

- More **reliable**
- **Easier to manage**
- And far more **scalable**

That's why platforms built from the ground up for the cloud are simply better suited for integration, automation, and long-term growth.

So, as I mentioned earlier, we put together a **scoring mechanism** for evaluating software **architecture**.

On the slide here, you'll see a matrix of criteria we use to assess each platform. These are objective criteria based on real-world use—not just marketing claims.

Talk to your IT team, or anyone experienced in this space, and they'll tell you—these categories *make sense*. This is how you evaluate ERP architecture.

Now, what do the **leaders** have in common?

- **Cloud-native architecture**
- Extensive **customization tools**—you can add custom fields to virtually any form
- Support for **multi-entity scaling**—great for companies with multiple divisions or subsidiaries
- A modern **relational database** back end—like Oracle, Microsoft SQL, or MySQL

Why does that matter?

Because the **database** determines how easily you can:

- Query data
- Build custom reports

- Integrate with other systems

A great example is **NetSuite**. It's cloud-native, deeply customizable, and runs on a robust back end. It checks all the boxes.

On the flip side, let's look at the **stagnated** group.

These are systems like:

- QuickBooks Desktop
- Sage 100, 200, 300, 400
- Microsoft Dynamics GP

Now, don't get me wrong—**GP** was a great product in its day. Back in the '90s and early 2000s, it was rock solid.

But it has **serious architectural limitations** today.

For example:

- It's a **"thick client"** application—meaning it has to be installed locally, on a server or desktop. It doesn't run natively in a browser.
- It doesn't scale easily across multiple servers or remote environments.
- It lacks a true modern API—and the one it *does* have is **proprietary**, which limits integration.

Bottom line:
These older architectures are not built for the way businesses operate today. And they're certainly not built for where businesses are going in the next 5–10 years.

So when we score these platforms, we're not pulling ideas out of thin air.
We're evaluating what **market leaders are already doing well**, and applying those best practices as the benchmark.

PRESENTER (STEVE)
So the next criteria we're going to talk about is **vendors and ecosystem**.

And to illustrate *why* this matters—whether we're talking about ERP, cars, or even phones—let me take you back to one of my personal experiences: **iPhone vs. Windows Phone**.

Now, I'll admit it… I'm a bit of a Microsoft bigot.
I carried a Windows Phone for *way* too long.
And truthfully? I still believe the **user interface** on the Windows Phone was better than the iPhone.

But here's why I had to stop using it:

It wasn't about the device. It wasn't about the OS.
It was the **vendor support** that went away.

One by one, major apps stopped supporting the Windows Phone.
My Chase app stopped working. My Marriott app disappeared.
And over time, the phone became… well, a brick.

I couldn't *do* the things I needed to do, even though the phone itself was fine.
Meanwhile, all those apps were continuing to work just fine on the iPhone.

That's the power of an **ecosystem**.
It doesn't matter how good your core product is if the rest of the ecosystem doesn't show up to support it.

Now—it's not all bad news for Microsoft.

Where Microsoft lost the vendor war in mobile, they've absolutely **won it in ERP**.

Their ERP platforms—Business Central, for example—have had **massive vendor support** for years. And this is where the strength of an ecosystem *really* shines.

So let's go back to those concentric circles I talked about earlier.
Microsoft focuses on building those **inner two layers**:

- Core capabilities
- Productivity capabilities

And then, they let **everyone else** innovate around the outside—adding in the **industry-specific** tools and custom workflows that make ERP platforms powerful and tailored.

How?
Through their **AppSource marketplace**, which allows:

- Hundreds of certified partners
- Thousands of extensions

- A dynamic, competitive space where third-party vendors push innovation forward

That's what makes Microsoft's ecosystem so strong.

Why does that matter for you?

Because when there's a robust ecosystem:

- **Innovation is constant**
- **Choice exists**
- **Pricing pressure stays in check**

Take **Procore** as an example.

Procore has its own ERP system. But its **project management module** is also available as an integration with:

- Microsoft Business Central
- Sage Intacct
- Other ERP platforms

That gives you, the customer, *leverage*.

If Microsoft or another ERP vendor suddenly triples their prices, you're not locked in. You can say, "Hey—I can take the parts I need and move them elsewhere." That **keeps everyone honest** and puts the **power in your hands**.

And beyond price protection, ecosystems also help you avoid **vendor lock-in**.

You're not forced to rely on a single vendor forever. You can build your tech stack in a modular way—swapping components as needed without having to blow up your entire ERP setup.

So when we look at ERP vendors that do this well?

- **Microsoft**, of course, is a standout.
- But also **Odoo**, the open-source ERP platform, deserves a mention.

Odoo takes it even further than Microsoft in some ways.
Not only can third parties write add-ons—they can even **modify the source code** itself.
That opens the door to deep, low-level customization when needed and really amplifies adaptability.

And when we talk about what makes a vendor ecosystem *strong*, here's what we look for:

1. **Partner certification programs** – with multiple tiers and strict requirements
2. **Validated customer references** – so vendors have to prove their value
3. A **competitive, open marketplace** – where the best ideas win

That's the kind of ecosystem that supports *you* as a buyer—today and for years to come.

PRESENTER (STEVE)
So what do I mean by a strong ecosystem?

Let's use an example close to home: **PaperTrl**.

When we develop an add-on for Microsoft's platform, that solution has to go through a **certification process** before it can be listed in the Microsoft marketplace. The same is true for QuickBooks Online and several other modern platforms.

Inside those marketplaces:

- Customer references are required for certification
- Commenting is enabled, so users can leave real, public feedback
- The platforms are transparent, and that transparency holds vendors accountable

That's the kind of structure that encourages **quality**.

Now, another thing market leaders do well:
They provide **ungated access** to both:

1. Their **API documentation**, and
2. **Test environments or sandbox versions** of their software

As a software provider, that's huge.

When PaperTrl is developing an integration, we need:

- Clear documentation that explains *how* their API works

- A test instance of their software so we can push through **test transactions**, simulate edge cases, and validate that everything works smoothly

The best ERP vendors **make this easy**—because they understand that when they empower developers, they're also improving the experience for their own customers.

A few vendors that do this particularly well:

- **Intuit**
- **Microsoft**
- **Odoo**
- **Oracle**
- **Xero**

These companies recognize that developers are part of the ecosystem—and they enable innovation rather than restricting it.

On the other hand, there are **vendors that don't do this well**.

They might have a marketplace—but they:

- **Restrict** who can access it
- **Limit** which vendors are allowed to list solutions
- Or enforce a "**pay-to-play**" model where vendors have to fork over 10%, 20%, or more of the revenue just to be part of the ecosystem

Sometimes the reason is simple:

Their **API isn't very good**.
It's hard to use, it's incomplete, or it lacks key functionality—so only a few selected partners are able to build anything meaningful.

Other times, it's more about control.
They'll say, "We want to control the quality of add-ons," but what that usually means is:

"We don't *have* a system in place to measure or maintain quality. We haven't invested in that infrastructure."

They don't have:

- Partner certification programs
- A public way to test and rate integrations
- Or online support for developers trying to build on their platform

And that's a red flag.

If you're evaluating ERP software, those are the **vendors to avoid**.
You do *not* want to end up with the ERP equivalent of the Windows Phone—great in theory, but with no ecosystem to back it up.

HOST (BRAYLEE)
Hey Steve, we have a question here from Joe Stevens.

PRESENTER (STEVE)
Sure—go ahead and read it.

HOST (BRAYLEE)
Joe asks:

"Is the order of the leaders on your slides significant? Meaning, do you rank them in that order?"

PRESENTER (STEVE)
Good question—and no, not necessarily.

We *do* have a scorecard at the end of the presentation where we talk about them in more detail. But the order on the slide isn't meant to imply ranking. It's more just a layout for discussion.

So the next piece we're going to talk about is the **API**.

Now, I know I've referenced this several times already without fully defining it upfront—so let's take a moment to do that.

API stands for **Application Programming Interface**.
And in the world of ERP systems, it's *the* component that **drives adaptability**.

Why does it matter?

Because the API is what allows:

- Data to move in and out of your accounting software
- Manual data entry to be eliminated
- Synchronization with other platforms
- Third-party tools to connect seamlessly

Basically, it's the engine behind integration.
Want to turn NetSuite into a construction-specific ERP? The API makes that possible.

A **well-designed API** sits on top of the accounting system and allows third-party apps to **replicate nearly every function** a human user could perform in the interface.
And I mean *every function*.

When that happens, it opens the door to:

- Business-specific automation
- Customized workflows
- Differentiation at the process level

So as we evaluate ERP APIs, what do we look for?

Traits of Leading APIs

Unsurprisingly, the **leaders in API capability** tend to be the same companies with the best **architectures** overall. Here's what sets them apart:

- **Standards-based design**: They use well-established, modern protocols like:
    - **JSON** – for message formatting
    - **REST** – for communication between systems
- **Public documentation**: Their API reference materials are published online, accessible to any developer.

- **Developer support**: They provide:
    - Help forums
    - Detailed documentation
    - Sample code
    - Dedicated developer teams to help with implementation

For example, we've written integrations to:

- **Visa's API**
- **U.S. Bank's API**

Both companies have teams that will work *for free* with developers like us to ensure successful integration.

This kind of **ecosystem support** is a huge factor in enabling high-quality integrations and driving innovation.

Pacing Systems

Then you have the **pacing vendors**—these are ERP platforms where *some* of the functionality is available through an API, but **not all**.

They might support a few transactions or limited modules, but you're constantly bumping up against walls where the API just doesn't cover the use cases you need.

Stagnated Systems

Finally, you have the **stagnated** API environments. These are the platforms where integration is:

- Difficult
- Proprietary
- Or locked down

Let me give you a few examples:

- **QuickBooks Desktop**

    - Not a REST API
    - Requires developers to write *QuickBooks-specific code*
    - That code can't be reused for other platforms like Microsoft GP
- **Sage 100, 200, 300, 400**
    - Import/export tools exist, but there's no **true real-time API**
    - That limits automation and responsiveness
- **Trimble (Viewpoint)** and **Deltek**
    - They claim to have APIs
    - But access is **restricted**—only approved partners can use them
    - Often, they'll say this is about "quality control," but in reality, it's about **market exclusivity**

Beware of that kind of restriction.

When ERP vendors say, "We only allow five partners to do payments," or "Only select vendors can do CRM," they're not protecting quality—they're protecting their own revenue streams and restricting competition.

The truth is this:

The **best software** emerges when **developers are free to compete**.

Let me give you a great example:
In the **Intuit (QuickBooks) marketplace**, if you're in construction, there are **20 or 30 different project management add-ons** available.

Out of those, two or three have risen to the top as **clear market leaders**—*not* because Intuit promoted them, and not because Intuit takes a cut of their revenue.

It's because:

- They're the best
- The **market decided**
- And Intuit simply enabled it by making the API accessible

That's how innovation should work.

Now, I do want to highlight one specific platform: **Blackbaud**.

If you'd asked me five or ten years ago, I would've put Blackbaud squarely in the "stagnated" category.

But today?
They've come **a long way**.

Blackbaud serves financial systems in **K–12 education**, **higher ed**, and **nonprofits**—and they've made tremendous progress in:

- Opening up their API
- Making their software more accessible
- Attracting third-party developers

And they've been a great company to work with.

PRESENTER (STEVE)
So, I mentioned earlier that we'd take a quick look at API documentation—and here's a good example.

What you're seeing here comes from **two different companies**:

- **Microsoft Dynamics Business Central**
- **QuickBooks Online**

This particular API is one that's **near and dear to my heart**.
Why? Because it's usually one of the *first things* we look for when a customer asks:

"Can PaperTrl integrate with [insert ERP system]?"

Specifically, we want to know:

- Can we **insert vendor payments** into that system?
- If PaperTrl makes a payment on the customer's behalf, can we record that back into the ERP to **close the loop**?

And in both of these examples, the answer is yes.

I just did a Google search for "Microsoft Dynamics Business Central create payments," and I landed right on the exact API call I needed. Same with QuickBooks Online.

These APIs allow our software to **replicate the action of a user sitting down and keying in a payment**—but it all happens programmatically.

The point is this:

This isn't black magic. It's not some exclusive club for tech insiders.

This kind of integration is possible—and **expected**—from the best ERP systems.
And it's something *every business* should look for when evaluating software.

Final Topic: Supply Chain Integration

Let's shift gears and talk about something else that's **near and dear to our hearts**:
Supply chain integration.

Not every company needs deep supply chain capabilities—but **many do**.
Some need full automation—from **purchase order** to **invoice** to **payment**.

Others just want a simpler way to **import vendor bills** or **sync supplier data**.

But here's the reality:
Not all ERP systems allow you to do this.

Now, think of the big guys—**Amazon**, **Home Depot**, **Walmart**.
For them, it's hard to tell where the ERP ends and the supply chain begins.
It's all one seamless, **integrated system**.

And that's where **industry-leading ERP platforms shine**—they make it easy for supply chain processes to flow through the system.

So who leads here?

Again—no surprise:

- **Microsoft Business Central**
- **QuickBooks Online**
- **Odoo**

These platforms provide:

- APIs for vendor data
- Support for inventory functions
- Real-time transaction processing

And they do it in a way that's accessible to third-party developers like us.

On the other end of the spectrum, there are **platforms that stagnate** here:

- No API support for B2B transactions
- Locked into **proprietary vendor portals**
- No direct support for inventory automation

And we've experienced this firsthand.

We've worked with ERP systems where we couldn't:

- Send in payment data
- Export or update vendor records
- Or even retrieve payment statuses

That makes **automation impossible**—and **adaptability non-existent**.

Wrapping Up: The Final Scorecard

So here's our final scorecard.
This is based on **objective criteria** across all the categories we've covered:
Architecture, API, Vendor Ecosystem, and Supply Chain Integration.

And as you might expect...

Leaders:

- **Odoo** – especially strong in openness and developer support
- **Microsoft Business Central** – great across the board
- **QuickBooks Online** – very solid, though we do ding them slightly on architecture because some API calls are *too bundled* and lack granularity

Strong Performers:

- **Sage Intacct** – does well, but slow on developer access to sandbox environments
- **NetSuite** – a power player with one of the earliest REST-based APIs

Pacing:

- **Blackbaud** – huge improvements in recent years, now offering a much more developer-friendly API and ecosystem

Lagging:

- **Deltek** – no open API, limited partner access, difficult to integrate
- **QuickBooks Desktop** – still widely used, and surprisingly strong in vendor ecosystem support, but weak in architecture and API capability.
  (Even Intuit knows this—that's why they're steering everyone to QuickBooks Online.)

PRESENTER (STEVE)
So just to summarize, here are a few key **takeaways** from today's session.

First, I hope everyone got something meaningful out of the presentation.

As I mentioned earlier—if you're looking to evaluate **base capabilities** of ERP systems (things like general ledger, reporting, and dashboards), I've provided several **external references** you can explore.

But more importantly, I hope I've helped **expand your thinking** around how to evaluate ERP software packages more **strategically**.

This is really about understanding:

- **Capability** vs. **Adaptability**
- What matters in the **short term** vs. what drives **long-term success**

You need both.
But too often, the focus stays on what looks good up front—when it's what's *under the hood* that determines whether your ERP is going to help or hinder your growth five years from now.

We need to move **beyond buzzwords**, beyond the **slick websites** and **glossy brochures**.

When you're in the evaluation process, ask harder questions:

- What **references** can you provide?
- What **add-ons** are available in your ecosystem?
- How easy is it to **import/export** data?
- Can I get a copy of your **system architecture**?
- Do you have an **open API**?

And make sure you're scoring these systems based on **structured criteria**, not just based on which one **demos the best**.

Because here's the truth:

Virtually *all* ERP software demos well.

You can literally walk into Barnes & Noble today and buy a book called *Demo to Win*. Every software company has trained their people on it. You can take the worst software out there and make it look good in a 45-minute demo.

That's why it's not about how polished the demo is—it's about the *infrastructure*, the *ecosystem*, and the *adaptability* behind the scenes.

And finally—think about the companies I referenced earlier:
Amazon. Home Depot. Walmart.
Or manufacturers that dominate their sectors.

For these companies, ERP isn't just a cost center—it's a **competitive advantage**.

I've been around long enough to remember some of their competitors:
Builder's Square, **Central Hardware**, and others who failed to keep up—who didn't treat ERP as strategically as they should have. And they're **not around anymore**.

Why?

Because they didn't invest in systems that were **connected**, **adaptable**, and built to **scale**.

So the right ERP system isn't just about accounting—it's about:

- Driving **revenue**
- Saving **money**

- And making sure **your growth isn’t limited** by your software

---

With that, I want to sincerely thank everyone for attending today.
I hope you got a lot out of the session.

If you have any follow-up questions or want to dive deeper, I’d love to connect.
My name is Steve Weber, and you can reach me directly at:

📧 **steve.w@papertrl.com**

Thank you all.