The Freedom to Run Your Business Your Way
Eliminate the Risk and Expense of Customizing ERP
The Freedom to Run Your Business Your Way
Eliminate the Risk and Expense of Customizing ERP

Customization is not a dirty word, but IT executives often rant about how it has made their department’s work an expensive and complex nightmare of patches, regression coding, and barriers to upgrading the software.

It was never intended to be that way. The whole notion of customization was born out of vendors’ desire to support a variety of business models. Every business is unique with particular IT needs. Businesses want the flexibility to tailor their systems to what they need.

Applications in most on-premise systems were developed for the lowest common denominator, addressing the functionality and features most common to the majority of customers. Many vendors claim to offer a flexible system with APIs that can extend applications or the customer can purchase middleware like Business Process Management (BPM) and integration solutions. These options aim to extend the functionality of an application beyond what’s in the box. The promise of an open, flexible system looks great, but many CIOs and IT executives don’t want the trouble that ensues when they try it.

As soon as a customer tailors the application to fit the business, things get complicated. Customizations are often done as needed and the result is piecemeal, rather than part of an integrated strategy. Over time, the application becomes more complex and difficult to manage. In extreme cases, the customer may have extended the application so much that the original underlying core is unrecognizable.

When upgrades become available, extensive testing and validation is required to keep the application from breaking down. This can cost millions of dollars and take six months for large enterprises.

This syndrome is known as “Frankenstein ERP.” It’s bolted on, ugly, and cobbled together out of various parts—and it behaves that way.

The result was one that vendors and customers never intended: systems became unmanageable. They were too costly and complex to maintain effectively. Upgrades became prohibitively expensive, and many organizations stayed with previous software versions, putting the latest innovations out of reach. Those barriers led to more bolt-ons and customizations, creating a self-perpetuating cycle of costly, complex, cumbersome systems for IT to deliver and maintain.

In short, the unintended consequence of this well-meaning strategy was that vendors were inadvertently giving customers enough rope to hang themselves—and many did. It’s a situation that’s all too familiar to many CIOs and IT executives.

The more critical issue may not be the lags in technology, but the time sink for the IT staff responsible for maintaining it.

The CIO of a successful medical-device company provides a great example. His company is not in the information technology business, he says, it is in the medical-technology business. His IT staff is dedicated to handling strategic challenges that advance the company mission, but that wasn’t always the case. IT used to spend much of its time patching legacy software, some of which goes back to the late 1970s and early 1980s. That was before the cloud and 20 years before the World Wide Web.
Configuration, Not Customization

Today’s cloud vendors face the same challenges as vendors of traditional on-premise applications. They must provide a system that’s flexible enough to meet customer’s needs, without adding complexity that slows down or paralyzes the business.

In moving from traditional systems to cloud-based systems, the responsibility to maintain and support applications shifts from the customer’s IT department to the vendor. That’s good for the customer, because it means that it can put IT resources toward more strategic functions. The vendor’s challenge is to create flexible systems that customers can tailor to particular business strategies and industry needs. But the flexibility should not introduce complexity that prevents the system from adapting as business needs evolve.

That effort is challenging for any vendor and for true Software-as-a-Service (SaaS) vendors it’s even more difficult. It’s particularly tricky for vendors that have a true multi-tenant system, in which all customers use a single version of a software product.

Workday chose a multi-tenant, SaaS-based approach and built a new system that overcame the legacy issues. While doing so, Workday had three possible options.

1. **Create a rigid system:** A use-it-as-is system could not be tailored for a specific business. Workday knew that businesses wouldn’t accept a system they couldn’t adapt.

2. **Create an open system:** When organizations have tried to customize systems freely and extensively, it has proved untenable. Many organizations have ended up with labyrinthine systems that are full of traps. From a vendor’s point of view, such systems are literally unsupportable.

3. **Find a happy medium:** A mixed approach would allow customers to tailor the system within limits. Enabling users to configure the system in a controlled way would protect future updates and allow support.

Even though customers could not make unrestricted changes, they could facilitate business processes and integrations within the platform.

Workday’s operating principle is configuration, not customization, so it chose the third approach. Customers would be able to manage their unique business needs without compromising on innovation.

The idea is to create a simple, adaptable foundation. It would use wizard-based screens that walk business users through the process. They would have greater autonomy, and IT would be freed of the burden of making patches and fixes and managing Help desk support. Structured business processes would be built into the product, eliminating the need for bolted-on software.

Some might argue that the third way is not true customization because of its restrictions. Workday, though, is comfortable with that, since customization brings little but trouble.

Configuration involves no coding. IT doesn’t have to make changes. In fact, business users such as HR, financial professionals, and administrative support staff do the configuring.

Business users make their modifications at the business-logic level, and the core data model remains off limits. Both vendors and customers avoid the curse of Frankencode and they can deploy upgrades without worrying about adding complexity.

Because Workday is cloud-based, it maintains the application. That way, the modifications made by customers don’t affect the core system and they work seamlessly across updates.

For context, consider Workday’s origins in the mid-2000s, when the Internet and the web were already mature technologies. Workday was starting with a clean slate. It could create an architecture using modern practices that had evolved out of enterprise resource planning (ERP) over the past 20-plus years and avoid the same mistakes.
Why Workday’s Flexibility Is Important

The business environment changes quickly, and even the pace of change continues to accelerate. Organizations need flexibility and they have to be agile to stay on top of market conditions, customer requirements, environmental conditions, and government and industry regulations. The more an organization can respond to, and even anticipate change, the more successful it will be.

Customer-initiated modifications have to be supported in a way that benefits both the vendor and the customer. The SaaS vendor can keep pace with its innovation goals, while the customer can count on the integrity of the application.

How Does Workday Do It?

Workday protects the core system. Customers do not have access to the back-end data model, nor do Workday developers. The core is where the most aggravating problems occur, the ones that make upgrades difficult, expensive, and complex.

But applications have to suit the individual needs of businesses. Workday offers customers the flexibility to configure the application to accommodate specific business requirements. These might be unique customer attributes or business processes. By doing this through configuration, not code, customers can make the system their own without altering the core of the application.

Customers know they can get the latest upgrades and innovations to their applications. This ultimately brings about the utmost adaptability. So the organization can easily respond to changing business requirements while keeping pace with the latest innovations.

It is a three-layer model.

- **The core**: The layers of data at the core are off limits. This protects the integrity of the code, which in turn protects the customer from making costly changes.

- **The business-logic layer**: Here, the customer can configure fields, reports, and business processes to suit the company’s needs. Typically, the person doing the configuration is a business user, not a tech user.

- **The user interface**: This is where employees enter the information and data that they work with.

Here are some examples of what Workday delivers, compared to legacy applications.

<table>
<thead>
<tr>
<th></th>
<th>Legacy Applications</th>
<th>Workday</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Organization Change</strong></td>
<td>Backend code changes</td>
<td>Configurable in the app</td>
</tr>
<tr>
<td><strong>Custom Fields</strong></td>
<td>Data model change</td>
<td>Configurable in the app</td>
</tr>
<tr>
<td><strong>Custom Labels</strong></td>
<td>Configurable by IT</td>
<td>Configurable in the app</td>
</tr>
<tr>
<td><strong>Custom Validations</strong></td>
<td>Backend code changes</td>
<td>Configurable in the app</td>
</tr>
<tr>
<td><strong>Calculated Fields</strong></td>
<td>Backend code changes</td>
<td>Configurable in the app</td>
</tr>
<tr>
<td><strong>Custom Reports</strong></td>
<td>Seperate BI infrastructure</td>
<td>Configurable in the app</td>
</tr>
<tr>
<td><strong>Business Processes</strong></td>
<td>Seperate BPM infrastructure</td>
<td>Configurable in the app</td>
</tr>
</tbody>
</table>
Organization. Workday enables customers to easily organize the system around their business, plan for change, and re-organize as they see fit. With no code involved, they simply fill in fields and use drag-and-drop in an intuitive graphic interface.

A business user can see what the organization looks like now and project it to some point in the future. You determine which elements to include and when changes will happen by moving selected elements on the chart around. You can review the model at any time. For example, you could create a change model that shows what the organization will look like five months from now, and with that information create the roadmap to get there.

Custom Fields. These fields allow users to configure the system to track data that is unique to their organization. Because it’s done without writing code, it does not change the underlying data model.

For example, a nurse administrator might want to track her department’s licenses. She can see when they are up for renewal, note the continuing education units (CEUs) that support license renewal, and schedule prompts to notify staffers.

Creating the field is very simple: under a worker object, simply define a set of new custom fields, and they will show up as configured. She easily creates custom fields that can then be used in reports, calculated fields, or other aspects of Workday.

The customer can specify the security policy that sets which groups can see which fields and configure it accordingly.

Custom Labels. These labels enable organizations to tailor Workday to align with their corporate culture. For instance, if an organization refers to its employees as “teammates,” custom labels allow you to re-label the term employee to teammate wherever that term shows up in the system.

Custom Validations. These provide organizations with a way to enforce better data quality on custom fields. You can easily apply validation rules to custom fields to ensure accuracy of the information. For example, if a return code must be entered when a return date is entered, you can configure a custom validation to enforce this.

Calculated Fields. These fields are one of the most heavily used elements within Workday systems. They are most used in reports and dashboards, but they can also be applied to business processes. You can access and apply them anywhere in the system.

Calculated fields allow a user to run a calculation of multiple fields, and then generate the result. They transform static data points by “activating” them, putting them in context, making them more relevant, and rendering them actionable. This function works well with Excel and with text. In fact, calculated fields can be used for more than data. You can combine text from multiple fields using calculated fields without coding.

Custom Reports. Traditional reports usually involve internal IT and a third-party business intelligence solution or reporting tool. But Workday’s custom reports give the business user autonomy and flexibility. Users can generate unique reports from within the application, without involving IT.

Users can define the report any way they like, and create reports and dashboards. They can change the look and feel of the template. They can expose it as a web service, creating a custom API. The best part is, they can take action immediately from within the report, narrowing the gap from insight to action.
**Business Processes.** These processes are core to the Workday system, which is based on a flexible business-process framework. It eliminates the need to bolt on a separate BPM solution.

Business users have traditionally had to collaborate with IT to model and implement their business processes. It often required several iterations to translate the business requirements correctly. With no need for coding, those closest to the business process configure processes in Workday. Business users can easily configure the process flow and associated rules using a simple tabular format. With Workday's built-in integration infrastructure, a user can extend the business processes to other systems outside of Workday without additional infrastructure.

**Alerts.** The system can be set to notify a user about whatever is important. The customer configures the system to create pertinent alerts and defines how those alerts should be delivered. Alerts can be defined for business-critical needs or personal business needs, such as being notified when expenses are approved.

**Workday REST API.** This allows organizations to create custom applications that involve Workday transactions, along with a new UI around Workday transactions to fit the organization's needs.

**Conclusion**

Workday’s configuration approach to business-process software accommodates unique business requirements much better than traditional on-premise applications. Because customers configure the system to match particular business needs, reconfiguring is much simpler than the old methods of coding, testing, and deploying.

Because configuration takes place at the business-logic level, the core data model remains untouched. This protects both the customer and the vendor. The customer can take advantage of vendor innovations without worrying about jeopardizing the future integrity of the system. The vendor continues to innovate, confident that advances will serve the markets.

Workday is responsible for maintaining the core application so that customer IT staff do not have to get involved. Traditionally, IT costs more than it earns, undermining company profits rather than contributing to them. If IT professionals spend more time advancing the organization’s mission, they do more than help trim expenses, they help drive revenue. As one customer CEO says, “Focusing on the bottom line only gets you so far.”

And for tech-intensive organizations, retaining strong IT talent is a challenge. Switching to a configuration model may help keep IT staff engaged and happy. Staffers who are working on mission-critical and strategic pursuits are less likely to be targets for recruiters than frustrated IT pros who are working on patches, fixes, and Help desk support.