

CUSTOMIZATION: A BLESSING OR A CURSE?

REMOVING THE BARRIERS TO INNOVATION AND CHANGE

Data Source

In this report, Mint Jutras references data collected from its 2016 Enterprise Solution Study, investigating goals, challenges and status and also benchmarked performance of software used to run a business

Over 450 responses have been collected so far (the survey is still open), from companies across a broad range of industries. This sample included responses from companies of all sizes, ranging from very small to very large enterprises.

Mint Jutras Definition of ERP

ERP is an integrated suite of modules that provides the operational and transactional system of record of your business. Because requirements across different types of businesses, the components required by an individual business will also vary, but today most ERP solutions extend beyond these minimum requirements.

If you want to achieve and sustain a competitive edge in today's global, digital economy, Enterprise Resource Planning (ERP) can be an important tool in your arsenal. If you define ERP as Mint Jutras does, as an integrated suite of modules that forms the operational and transactional system of record for your business, every business needs it. Yet in spite of the cost and effort invested in ERP, few realize its full potential. While there are many obstacles encountered along the way, according to the 2016 Mint Jutras Enterprise Solution Study, the top-rated challenge to achieving the goals of ERP is related to customizations.

*In a world where people and enterprises alike want things "their way," it is easy to understand why so many companies have customized solutions. What is not so easy to understand is why these customizations still cause so many headaches. Next generation ERP **should** allow you to personalize, configure and extend solutions without creating obstacles to innovation and realization of goals. Clearly, we have a disconnect between possibility and reality. If customizations are holding you back from achieving the full potential of your solutions, perhaps you are going about it all wrong, or maybe you just need a new ERP.*

WHY CUSTOMIZE? YES, IT MATTERS

Before going any further, it is perhaps wise to question the motivation behind customizing the software that runs your business, because the "why" is often just as important as the "how." In the early days of ERP these customizations often resulted from limitations of the software, resistance to change, or some combination of both. This set a precedent that continues today, sometimes resulting in changes that simply may not be necessary.

The depth and breadth of early ERP was very limited compared to the ever-expanding footprint of solutions today, often leaving functional gaps that were filled with customizations or other point solutions. Even if a complementary solution was purchased to fill the gap, more often than not integration with ERP meant custom code.

Furthermore, business processes built into early ERP solutions weren't necessarily the most intuitive or the most efficient. Even with lots of training, these systems didn't work exactly the way people worked, which meant the same ERP that was supposed to make life easier, sometimes made it harder.

Fast Facts

- ✓ *The top challenge this year preventing companies from achieving the goals set for ERP was "Customization related challenges"*
- ✓ *61% of companies say customizations are somewhat to very likely to prevent them from upgrading*

Even where features and functions were available, the look and feel, as well as the workflow dictated by ERP was rigid and drew many complaints. These complaints often resulted in customization.

Unfortunately customization also used to mean mucking around in source code, which built barriers to moving forward with updates and upgrades. That was because in the past all the logic was "programmed" into that source code. This made business applications like ERP rigid and inflexible. Sure, there were always some configuration options, but those options were constrained by the logic embedded in the source code.

Therefore, once you heavily customized the code, there was an excellent chance of being stuck. Even as your maintenance dollars were funding your solution provider's innovation, you were never able to take full advantage of upgrades. Even if your solution provider has been devoted to filling those functional gaps, have you been just as devoted to eliminating customizations wherever and whenever possible in order to consume that innovation?

If not, then perhaps your customizations were less motivated by functional gaps and more motivated by a "we've always done it that way" mentality. This could be pervasive throughout the company, or the result of one or more individuals who were (are?) resistant to change. Years later you might not even be able to explain why that customization was needed. Maybe Edith in the receiving department refused to adapt to the new system. Then again, maybe Edith retired five years ago.

Or perhaps you never eliminated those customizations simply because it took so much time, effort and money to put them in. Once in, you figured, if it's not broken, why fix it? But what else did you sacrifice because of them?

GOOD REASONS AND BAD REASONS

There are many good reasons to customize solutions, but equally as many bad reasons. Customizing a solution to provide a clear differentiation in your market is a good reason. Requiring a new system to operate exactly the same as an old system, or worse yet, no system at all, is a bad reason. Why bother with a new one?

Personalizing the user interface for your company, to roles within your organization, or even to the needs of individuals are all good reasons to customize, particularly if it encourages engagement with the system. In those early days, a small percentage of employees in any company actually put their hands directly on ERP. You had your heads-down data entry folks and a few super users. Those days should be long gone, and if you are still operating like this, you are indeed stuck in the past. Today on average 64% of employees have direct access to ERP. The last thing you want is to have them spend more time trying to work around the system than with it.

Today on average 64% of employees have direct access to ERP. The last thing you want is to have them spend more time trying to work around the system than with it.

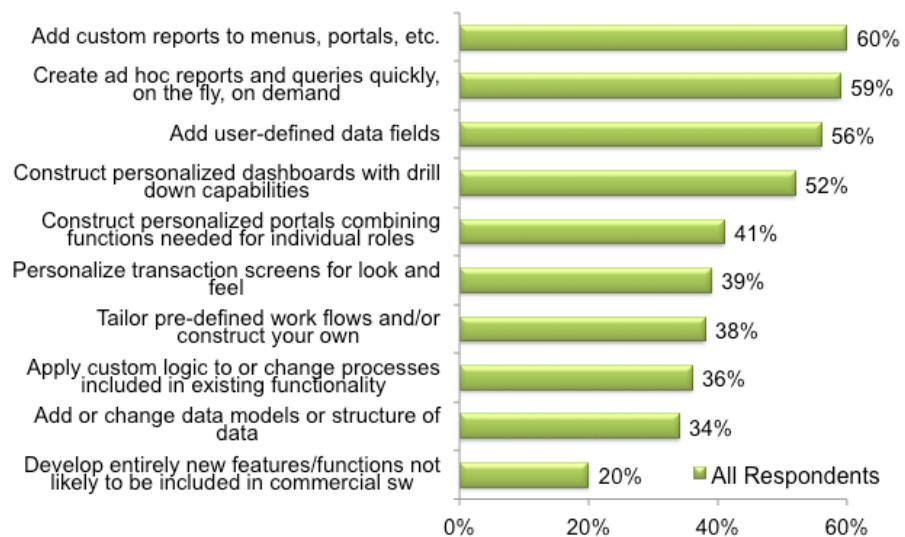
And that interaction with ERP should be far less labor-intensive today than it ever used to be. While user interfaces have become more intuitive and visually appealing, the best user interface is often no user interface. Steps can be automated, and data can be collected from any number of different sources, ranging from devices (machines, equipment, things) to the Internet. Workflows can (and should) be customized to your business processes, governed by business rules that can be easily changed as business changes. It should be just as easy to get data out of ERP than it is to get it in – even if you can’t anticipate all your data needs up front.

In a sense, all these “good” reasons result in “customizing” ERP, but not in the same sense as before. There should be no invasive code changes required. In fact, while historically information technology (IT) staff and developers were intimately involved, in many of the most modern ERP solutions today, non-technical business users and decision-makers can make these changes. So let’s not even call them customizations. Think instead of personalization, configuration and extensibility.

HOW DO YOU CUSTOMIZE? LET ME COUNT THE WAYS

Keeping in mind that most any change to the software or the user experience is typically referred to as “customization,” we asked our 2016 Enterprise Solution Study participants what level of customization they **believe they need**. Respondents were allowed to select as many as needed from a lengthy list (Figure 1).

Figure 1: What level of customization do you believe you need?



Source: Mint Jutras 2016 Enterprise Solution Study

“Customization”

Many modern, technology-enabled ERP solutions today deliver a high level of personalization and configuration without customization as defined in the classic sense of invasive code changes.

Figure 1 is sorted by the popularity of these different types of customization, but all of these easily fall into one or more of the following categories:

- Reporting and inquiry
- Look and feel of the user interface
- User-defined data fields and structures
- Logic, features, functions, including workflow

The bulleted list above is sequenced (from least to most) in terms of likelihood these software changes would/should result in actual customization in the classic sense of the term, requiring invasive code changes.

REPORTING AND INQUIRY

Early ERP solutions were famous (or perhaps infamous) for being much harder to get data out of than into. First of all, it is very hard for solution providers to anticipate the exact reporting needs of any customer, much less all customers. This is definitely an area where one size does not fit all. And let's face it; those reporting requirements can change over time.

Today it is less important to judge an ERP solution by the depth and breadth of reporting than it is to judge it by the ease with which existing reports can be modified and new reports added. While this task used to sit squarely on the shoulders of the IT department, next generation solutions put these capabilities directly in the hands of the user. But for this to be successful, the process cannot intimidate business users. They need to be able to easily locate data, drag it and drop it into a format that is easily read and interpreted. Being visually appealing is also a plus, and that often means pretty pictures (a.k.a. charts and graphs).

Each decisionmaker needs their own personalized dashboard and a portal that is truly a gateway to data that will answer their questions today and tomorrow. And it is no longer enough to rely on laptops and desktops or worse yet, subordinates to do the digging. Answers need to be at their fingertips, available on any mobile device they happen to carry.

If your business leaders are still using smart phones as dumb phones to call someone else for answers, you are stuck in the past. If you still have to wait in line to get your IT department to put a new field on a report, change the sorting and selection, or the formatting, or even create an entirely new report, it is important to know that with today's solutions, it doesn't have to be that way. That alone might be a hint your existing solution may be more of an inhibitor than an enabler.

LOOK AND FEEL OF THE USER INTERFACE

Reporting, inquiry, dashboards and portals are also related to the overall look and feel of the solution and that translates to the user interface. This look and feel used to be built into software. But for years now, ERP has been separated

Today it is less important to judge an ERP solution by the depth and breadth of reporting than it is to judge it by the ease with which existing reports can be modified and new reports added.

If you still have to wait in line to get your IT department to put a new field on a report, change the sorting and selection, or the formatting, or even create an entirely new report, it is important to know that with today's solutions, it doesn't have to be that way.

into layers and the first layer to be removed from the guts of ERP was the user interface. Over the years, solution providers had a lot of incentive to do this, even apart from making users happy. As more and more companies expanded globally, a user interface that could easily be translated into multiple languages became table stakes. The easiest way to insure speed and quality of that translation was to extract the bits that needed to be translated from the logic of the software. In doing that, the user interface became infinitely more customizable without ever touching source code. Changing field names was as simple as translating them.

Of course this was just the first step. Once companies saw how easy this was, they wanted more. They wanted to move things around on the screen, hide data and maybe even add more. They wanted to make full use of a large screen or make the most out of the limited real estate on the screens of their mobile devices. Extracting the user interface from the underlying source code doesn't necessarily mean a non-technical business user can accomplish all this. If you can't do this with your current system, you should know that new, technology-enabled solutions can... safely and securely.

There have always been a few user-defined fields that could be tagged with a label and used as needed by the customer. The problem was (and still is for older solutions) that's about all you could do with them.

DATA FIELDS AND STRUCTURES

The need to add user-defined data fields placed in the top three types of customization required. This option is actually quite common even in older, legacy solutions. There have always been a few of these fields that could be tagged with a label and used as needed by the customer. The problem was (and still is for older solutions) that's about all you could do with them.

In addition, they were typically attached to a particular record in a particular file. You might have user-defined fields in the part master or the customer record or maybe the vendor master file, but perhaps you need a user-defined field for a supplier part number. This will vary by each of the possible suppliers for the part, so you can't associate it with the part itself. And you can't put it in the supplier master record because it will vary by part. So unless there is a user-defined field in the file that associates multiple vendors for a part, you have no place to put your supplier part number and the customization becomes much more disruptive than re-labeling a field.

Over time, solution providers have been adding more flexibility for user-defined data, but do your due diligence. Determine any limitations of these user-defined fields and also figure out what you can (and can't) actually do with them once you have them. Can you build any logic around them or apply any business rules? Does this involve source code changes?

Of course sometimes the requirement for user-defined data can extend beyond adding a single field. Take our example of the supplier part number. While newer solutions are very likely to support linking multiple possible suppliers to a purchased part, older solutions or even brand new, modern solutions (still under development) might not. In this case you don't simply

Many solutions today are built on “platforms” that allow developers to create data structures and software without the complexity of building the infrastructure typically associated with developing an enterprise application.

need a user-defined field, you need a new data structure. Our example is a relatively simple one, and adding or changing the data model or the structure of the data can be far more complicated than this. Investigate carefully the difficulty involved.

To better understand the opportunities, as well as the limitations, you will most likely need to have some level of understanding of how the solution is built. Many today are built on “platforms” that allow developers to create data structures and software without the complexity of building the infrastructure typically associated with developing an enterprise application. You will need some technical expertise in evaluating these platforms. If you don’t have that expertise in house, there are plenty of consultants available to help. Clearly developers benefit from using the services delivered with a platform, speeding the development process, but this can also translate to benefits to the business, helping companies keep up with the accelerating pace of change.

Which brings us to the final category of customization and the one most likely to involve potentially invasive and disruptive code changes. As important as the user interface is today, it is important to look at the overall user experience. That means not only how it looks and feels, but also what it does for you.

LOGIC, FEATURES AND FUNCTIONS

What do you do when the software doesn’t exactly match your business? This may or may not involve entirely new features and functions. More often than not, this mismatch is simply the result of the workflow not being aligned completely with your processes. Early ERP solutions largely embedded the sequence of events into the code, perhaps with some switch settings added to configure certain steps within the processes.

Modern systems are more likely to define the flow of work separately. Think of it as a separate layer, much like the user interface layer. Not only can the flow be configured, but it can also be governed by a set of business rules.

Modern systems are more likely to define the flow of work separately. Think of it as a separate layer, much like the user interface layer. Not only can the flow be configured, but it can also be governed by a set of business rules. These rules might be used to determine behavior of a function or to configure next steps in a workflow. Business rules might define different thresholds for approval (e.g. all purchase orders require approval but those over a certain value require an extra step in the approval process). To be most flexible and adaptive, these business rules should not be buried in the source code. Look for the equivalent of a rules engine that allows non-technical users to manage and maintain them.

These business rules might also be used to trigger alerts, notifying managers when events occur (e.g. a big order comes in) or when they fail to occur (a scheduled delivery date is missed). With the “right” system, workflows can be configured extensively without ever causing you to “customize” in the traditional sense. And if you need to further tailor business processes that are quite universal across all businesses (e.g. processing payments, collecting cash,

Are you changing the software simply to perpetuate a process that is different, but not better? If the change does not add specific value or somehow differentiate you in your market, don't do it.

Don't "customize" your ERP; extend it.

issuing purchase orders, receiving materials, etc.), you might want to circle back to the original question: Why?

Are you changing the software simply to perpetuate a process that is different, but not better? If the change does not add specific value or somehow differentiate you in your market, don't do it. If you have already made invasive code changes that are limiting your ability to move forward, innovate and grow, remove them and take better advantage of updates your existing vendor has been delivering. Of course, if this is a very substantial effort, you might find implementing an entirely new, modern, configurable solution is actually the better and easier route to take.

While the need for this type of workflow configuration is very common, it may not suffice. About one in five of our survey respondents (20%) expressed the need to develop entirely new features or functions not likely to be included in commercially available software. This is the requirement most likely to require you to develop your own software, making the choice of development platform very important.

An added word of caution though... don't "customize" your ERP; extend it. Develop these new features and functions and integrate them where needed, but in a way that is more loosely coupled than tightly embedded. What's the difference, and more importantly, what is the advantage of loosely coupling?

Traditional ERP was developed as a tightly integrated set of modules. Not only do all modules of an ERP solution share a common database, but all are developed using the same tools and technology and they all move forward in lock step. This eliminates data redundancy and any need for separate integration efforts. But it also means all the parts need to move forward in lock step. If you apply the same concept to a new "custom" function, you have the same problem you have always had with invasive source code changes. You can't upgrade to the latest release without taking a serious look at that custom-developed code.

Instead, develop your new function as a separate component and loosely couple it to ERP. That way, if you wanted to replace it, you would just have to uncouple it and swap in a new one – sort of like uncoupling one of the cars on a train. It just takes a standard coupling, right? Of course it is a little more complicated than that, but that's the general idea.

It is beyond our scope here to fully detail the difference between a loosely coupled extension and customization embedded in your ERP. Suffice to say it will likely involve object-oriented development and services oriented architecture. If you are running on older, perhaps proprietary architectures that prohibit you from taking advantage of these newer technologies... that again might be sufficient reason to consider replacing what you have today.

CONCLUSION AND RECOMMENDATIONS

In your company, is customization a blessing or a curse that prevents you from realizing the full potential of your ERP? Customizations that build barriers to innovation and growth are a curse. If your customizations prevent you from

- consuming the innovation provided by your solution provider
- adapting to business change
- seeking new business and new business models

... they are a curse.

But customizations can be a blessing, if they help your employees engage with ERP, allow you to address your real business needs and/or provide you with differentiation in your market. Generally speaking, customizations that are blessings aren't customizations at all in the traditional sense. You need to redefine customization. They cannot require invasive code changes that inhibit rather than enable your evolution. Next generation ERP allows you to personalize, configure and extend solutions without creating obstacles to innovation and realization of goals. If your customizations are holding you back, perhaps it is time to re-evaluate them. Perhaps it is also time for a new ERP.

About the author: *Cindy Jutras is a widely recognized expert in analyzing the impact of enterprise applications on business performance. Utilizing over 40 years of corporate experience and specific expertise in manufacturing, supply chain, customer service and business performance management, Cindy has spent the past 10 years benchmarking the performance of software solutions in the context of the business benefits of technology. In 2011 Cindy founded Mint Jutras LLC (www.mintjutras.com), specializing in analyzing and communicating the business value enterprise applications bring to the enterprise.*