



WHITE PAPER

Flexibility will reduce the hidden costs behind ERP software

Casey Lawrence, Enterprise Software Specialist

ABSTRACT

This document explores the hidden cost of customizing enterprise software along with an understanding why customizations will most likely produce the largest hidden cost during implementations. Included is an explanation how to reduce customizations by selecting enterprise software with flexibility. Explained are the different types of flexibility to look for in enterprise software and how to use this flexibility to minimize the hidden costs when implementing enterprise software.

FLEXIBLE VERSUS RIGID SOFTWARE

The word “soft” in software is a misnomer. Most enterprise software is really far from soft and flexible. As a matter of fact, it’s about as flexible as a concrete driveway. Enterprise software includes ERP, SCM, CRM, WMS, PLM, etc. for all industries.

Flexibility is an important design concept. There is strength in flexibility. Any engineer understands the principle of flexibility in design. If a structure is properly designed with flexibility, it will last longer and withstand more stress. Software is the same. If a software product is designed to be flexible, it will become strong, powerful and very useful.

If you study the history of software over the past 30 years, you will notice most of the really successful software products were flexible. Most of these successful products are still around and thriving today. For example, consider the spreadsheet—now that was a great product. So why is the spreadsheet so successful? Because it is extremely simple and flexible. The spreadsheet has the capability to perform multiple tasks, including interactive business intelligence, reports, financial statements, analysis and forms, all within one user interface. Other examples of successful products include word processors, desktop publishing, web browsers and multimedia authoring software such as PowerPoint®. When simplicity and flexibility meet software, you get an explosion of success. However, enterprise software is not simple and it is far from flexible. A flexible product is one that can be altered by a user with few technical skills, rather than by a full-on programming effort by an experienced developer. The spreadsheet would have died on the vine long ago if you

had to hire a programmer to develop each spreadsheet change.

Oddly, flexibility is not as highly valued in the enterprise software market as it is in other software markets, which is probably due to the fact that rigid software is all we know and is all that is available in the enterprise software sphere. And you can blame current development methods for this rigidity. This is because one or more forms must be developed to address a single business process. A form is a computer window or frame used to view and/or manage data. It is rare in the enterprise software industry to design a form to address multiple business processes, like sales orders and purchase orders, even though the navigation might be the same. This is because they are considered completely different business processes. Therefore, the path is forever divided between these two forms because they involve different business processes. Eventually this leads to varying methods of navigation and functionality between the two forms. It’s just the nature of enterprise software development. Once you are on a business process path for a form (or set of forms); the navigation and functionality will most likely vary significantly from other business process forms. For example, a form will be developed to specifically enter and maintain sales orders. However, the entry and maintenance of sales orders varies significantly between companies. For example, entering a sales order for an automobile is significantly different from entering a sales order for a toy. Therefore, an enterprise software supplier must continually tinker with the form and add layers of functionality to address all possible sales order scenarios for all industries. After years of alterations and additions, end result is a complicated, confusing and rigid set of forms for each business process. It’s just how the industry operates.

Before discussing flexibility in enterprise software (or the lack thereof), it is important to understand the historical strategies used by software suppliers when addressing all business activities, and why these strategies are failing. Since enterprise software falls short of flexibility, other solutions were needed in order to hide the gaping hole of flexibility.

To begin with, software suppliers went to work developing forms and processes to address as many

business activities as possible. By the late 1980s, enterprise software began to expand and some suppliers began to realize a major dilemma. That is, it is not possible to develop forms and processes for every possible business activity in the future. A new approach was needed.

One solution to the problem began in the early 1990s. The solution was to identify and standardize all business processes. This direction sent the industry further down the path of rigidity and further away from flexibility. Enterprise software suppliers hoped they could force companies to comply with standard business processes, called "best practices." The marketing machine went into gear, convincing companies to comply with a software supplier's pre-defined business processes, claiming the software supplier knew the best way for companies to operate.

However, by the late 1990s, it became apparent there were just too many variations to business activities. Companies began to push back from the notion of complying with pre-defined business processes. Companies that valued their unique business processes did not want to comply with pre-defined processes for fear of dulling their competitive edge. Therefore, the approach to force companies to comply with rigid business processes started to lose momentum. But this left the industry in a real pickle because a solution didn't really exist. Without true flexibility within enterprise software, how does a company select and implement an enterprise software product and make it work for their specific and unique requirements? The only solution is to customize.

Customizations are the largest hidden cost behind nearly all enterprise software implementations today. There are more horror stories regarding customizations than any other facet of enterprise software implementation.

To clarify, there are basically two different types of customizations. There are customizations to *change existing* software forms, processes or reports. And there are customizations to *add new* forms, processes or reports. The reason for customizations is to provide additional functionality in order to address the gap between a company's requirements and the generic software functionality. Customizations which alter existing forms, processes or reports are usually frowned upon, whereas customizations adding new forms, processes or reports are acceptable. Most suppliers will not support their software if there are customizations to existing forms, processes or reports. This is understandable because software suppliers have enough difficulty supporting their own product, let alone any customer changes. Therefore, the rule is "hands off" when it comes to customizations on existing forms, processes or reports. On the other hand, new customizations are acceptable, and will be the best way

to bend a rigid software product to a company's specific needs.

I should mention here, there are ways to alter forms, processes and reports within enterprise software and still get support, but that discussion is beyond the scope of this document.

Now we arrive at the core of the dilemma with most enterprise software. That is, how do I select and implement an enterprise software product and make it work for my specific needs, without a lot of hidden costs involving customizations? This leaves companies with three options. These are:

1. **Implement generic enterprise software:** Implement the generic version of an enterprise software product "as is," without any customizations.
2. **Minimize customizations:** Accept the need to customize, but with a minimal number of customizations, especially to existing forms, processes and reports.
3. **Look for flexibility:** Select enterprise software with some flexibility functionality, and utilize this flexibility to further minimize or eliminate customizations.

The following sections elaborate on the pros and cons for each of these options.

IMPLEMENT GENERIC ENTERPRISE SOFTWARE

One option to reduce hidden costs is to avoid customizations altogether by implementing generic enterprise software functionality. This is the old "comply" approach which, believe it or not, is still rather prevalent in the industry.

With a generic implementation approach, a company must comply with all pre-defined business processes within a product. This translates into no changes or additions. Users are forced to use the new business processes without any unique functionality.

Some small companies might be able to get away with implementing generic enterprise software without any customizations. If a company's requirements fit the standard enterprise software model, this approach might be an attractive option. But it is very difficult to avoid new customizations, especially reports, even for small companies with standard business processes.

Unfortunately, most generic enterprise software implementations end up changing midstream and taking on a significant number of customizations. The reason why companies fall into this trap is primarily due to poor

planning based on marketing hype. The primary source of the marketing hype comes from either the software supplier or consulting company. This is done to sell the product or consulting service. The trap is to get in the door with a simple and inexpensive solution.

Unfortunately, naïve owners and managers are “sold” on this approach during the sales cycle and end up getting duped later when unique and critical business processes are uncovered during the implementation.

For most medium and large companies, implementing a generic enterprise software product without customizations is a myth. There will always be a gap between company requirements and software functionality. Some of these gaps will be critical to the success of a company and must be addressed. Therefore, customizations will have to be considered when implementing enterprise software.

Starting a generic implementation and later switching to allow customizations will be more expensive than just accepting and planning on customizations initially. You don't want to start building a freeway and discover midstream you forgot to install a new water line. It will be very expensive to tear up the freeway and install the water line. To stop, redefine and re-start an implementation project is very expensive. Paying for functional consultants to wait while new customizations are defined and developed is very expensive. Added to this expense is the cost of bringing in new technical consultants to address unexpected customizations.

If you are a small company with a simple and standard set of requirements, and feel perfectly comfortable that there are no gaps between your requirements and the software functionality, then by all means do a generic implementation. Otherwise, avoid generic implementations like the plague.

MINIMIZE CUSTOMIZATIONS

Another option to reduce the hidden costs behind customizations is to just accept the fact that *you must customize*; but to keep hidden costs down, you must minimize customizations. A strategy should be put in place to minimize customizations during the selection and implementation phases.

Most enterprise software selection and implementations attempt to put in place a strategy to minimize customizations. And unfortunately, nearly all implementations end up exceeding the budget and expectations for customizations. It is rare to actually meet the budget and expectations regarding customizations. During most implementations, the customization budget will go beyond expected plans, resulting in project delays and excessive surprise costs.

Not only are customizations the largest portion of software implementation costs, but the cost to customize is compounded when new customization needs are uncovered during an implementation. Changes or delays to an implementation plan are expensive and the reasons for most delays are due to new customizations.

Customizations will result in two kinds of hidden costs. First: the cost to develop customizations. Second: the cost to maintain customizations. All customizations must be reviewed, tested and adjusted during the implementation, and again for each software upgrade thereafter, which usually occurs once a year. I have seen customizations cost a company four times the initial cost of the enterprise software product. Out of the 50 plus companies I have implemented, none of these escaped new customizations and only a few were able to avoid customizations to current forms, processes and reports. Therefore, most companies will go over budget on customizations. The hidden cost of customizations is the most painful and expensive part of nearly all implementations.

The reasons for these unexpected costs are primarily due to the failure to identify gaps between software functionality and company requirements prior to the procurement of an enterprise software product, and sometimes during the implementation planning phase, where often the best strategies to identify and control customizations end up failing.

You would think a reasonable solution to the problem would be to identify all possible gaps between requirements and generic software functionality prior to a purchasing decision. However, this solution has a dark side. Trying to identify all gaps prior to a purchasing decision will most likely end in frustration when you get locked into over-analyzing the situation. Eventually, the process might turn into gridlock. It's like a decision to get an army moving without thoroughly sweeping every square foot of the terrain for land mines. You must get moving. Therefore, some risks will be required. Someone is going to get hurt. The same is true with enterprise software selection and implementation planning—at some point you must leave the details and get moving. And, this is why so many companies step on the customization and enhancement land mines.

Another reasonable solution to the problem of minimizing customizations is to just focus on major gaps instead of all gaps between requirements and software functionality prior to making a purchasing decision. Unfortunately, this solution has its own problems. The problem is identifying what is considered as a “major gap.” What might be a major gap with one software product might be minor to another. Each enterprise software product will vary with business process depth. The size of the gap depends on the amount and type of programming required to address a particular

requirement. And there could be hundreds of requirements. So which requirement will result in a major gap? You will not know unless you have each supplier extend each requirement with a solution and a cost. Therefore, you are back to reviewing all requirements. As you can see, this is going to be difficult. Neither you nor the supplier is going to do this amount of detailed work with all enterprise software products being considered.

Now we have returned full circle, that is, confused about customizations. So, how do you get a grip on the hidden cost of customizations? All you can do is size up the situation the best you can and get going. It's like wading into a black swamp—you don't know how deep it is until you make the next step, which might take you in over your head. This is why so many companies end up with enterprise software products from "tier-one" suppliers, hoping the customizations will be minimized due to the size and depth of the product.

I know what you are thinking. You are saying—wait a minute, if it is silly to implement generic enterprise software and it is dangerous to customize and enhance an enterprise software product, what should a company do? The only hope is to combine software flexibility with a strategy to minimize customizations.

LOOK FOR FLEXIBILITY

To understand the need for software flexibility, it is important to understand why it is difficult to define and maintain a customization minimization strategy.

the middle is nearly impossible and requires a lot of skill in holding your ground with users and management. This is what happens to nearly all implementations. You start off either trying to perform a generic implementation or a strategy to minimize customizations and you end up heading towards unlimited customizations, and the hidden costs go through the roof. Implementation management must hold their ground, but often the pressure becomes overwhelming, especially when critical missing functionality is uncovered during the implementation.

To reduce the risk of stepping in over your head when wading into the swamp, you must minimize your risk through a customization strategy using flexible enterprise software functionality. That is, the ability to customize the software without heavy programming. What I mean by "heavy programming" is when you must bring in a developer with years of experience to make changes and additions. Minimizing the cost and effort behind customizations is very difficult without some software flexibility. Software flexibility will provide a means to minimize customizations and provide a balanced strategy towards customizations.

The hidden cost behind customizations is the cost of programming. If you find a way to simplify or eliminate programming, you will significantly reduce the hidden costs. This is what true flexibility delivers—the means to get a handle on outrageously expensive customizations.

Flexibility—combined with a good strategy to identify as many customizations as possible early in the game—should give a company the necessary tools to keep the

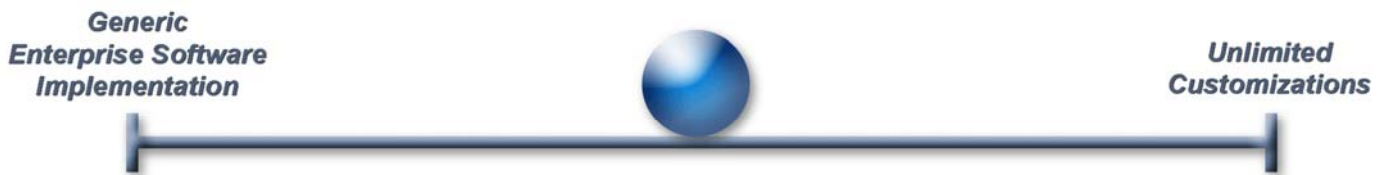


Figure 1: Balancing Customizations

Figure 1 illustrates how difficult it is to make a balanced decision regarding customizations. The illustration shows a marble on a stick. If you hold each end with both of your hands and try to keep the marble in the middle, you will understand how difficult it is to find a balanced approach to customizations. It is easy to let the marble roll to the left and go the direction of generic enterprise software implementation, which has the illusion of simplicity. Or, roll to the right and fall into the unlimited customization trap. Because it is so difficult to limit customizations, once you start it's hard to stop. Both sides are easy to slide into. But keeping the marble in

marble in the middle.

However, before you get too excited, you should know that flexibility is a missing or elusive feature in most enterprise software. As a matter of fact, true flexibility within enterprise software doesn't really exist today. Yes, I know, this is discouraging. But, there is hope. The good news is that some enterprise software products have elements of flexibility, which I recommend looking for when selecting a product.

The following is a short list of flexible functionality to look for when selecting an enterprise software product:

- **Add custom columns to forms:** Look for the capability to add new columns to an existing form using a simple user-friendly process. For example, you might want to add some columns to the customer form in order to maintain unique data about a customer. Also, make sure the new columns are part of the form and associated table for reporting. Be aware, most products will limit the number of user-defined columns. Also, an added bonus will be the capability to define drop-down lists, formatted data, validations and events tied to the definition of these new columns. This will reduce the need of customizing existing forms or creating new forms.
- **Change form layouts:** Look for the capability to alter a form's layout through a user-friendly point-and-click process without programming. This includes hiding, un-hiding, moving and re-sizing columns. Most enterprise applications have a large number of columns in order to satisfy all company requirements. Excessive columns will make the forms complicated, rigid and time-consuming for the users. If columns can be hidden, moved or re-sized, you will improve user productivity and reduce the need to customize existing forms or create new forms.
- **Change column sorting on forms:** Look for the capability to change column sorting on a form using a simple user-friendly, point-and-click process. This will eliminate the need to create custom inquiries with different sorting themes.
- **Toggle between single and multi-record layouts on forms:** Look for the capability to toggle between a single-record layout and a multi-record scrolling layout on a form. Sometimes it is important to see all of the columns for a single record on a form, and sometimes it is important to see multiple records on a form. If both of these layouts are provided within the same form through a toggle while on a record, the need to create additional custom inquiries and maintenance forms will be reduced.
- **Change column validations on forms:** Look for the capability to add additional validation logic for one or more columns when specific activities are performed. This might require a little programming, but the process should be as simple as possible, helping you avoid the need to customize forms or add complicated custom workflows, alerts or triggers to the database.
- **Define security on forms:** Look for the capability to allow record additions, changes or deletions; as well as the definition to view and access certain columns based on a user type or a specific user. This is the most requested and common customization during any implementation. If this feature is available, you will definitely reduce customizations, because most customizations involve security issues.
- **Add events on forms:** Look for the capability to add event triggers on forms. An event is a process which will launch when a certain action or combination of actions occurs on a form. The event is usually a custom program to perform some related activity, like sending an email notification. This capability will help you avoid the need to customize forms or add complicated custom alerts or triggers to the database.
- **Save form personalizations:** Look for the capability to save all form changes (mentioned above) as a personalized form name. The personalized form may be defined to be the default form for all users, specific users or user types. This way, these adjusted forms may be created, saved and linked to specific users, significantly reducing customizations.
- **Flexibility works with upgrades:** Make sure all of the flexibility suggested in this list remains in place during upgrades. All of this great flexibility is useless unless it is automatically part of an upgrade. If form adjustments must be redefined after each upgrade, it is not considered flexibility.
- **Zoom functionality:** Look for the capability to connect (or zoom) to related data while on a specific record or column. For example, if I am on an item record, I might want to zoom to all related purchase orders, sales orders, inventory, etc. related to the item I am on. This feature will both improve user productivity and reduce the need for customization links to existing forms or menus.
- **Simple addition of new forms:** Every company has unique data which must be maintained. Therefore, look for the capability to easily define new tables and forms. I have seen some products with user-friendly functionality to define and build new tables and forms. However, most products accomplish this through programming. If programming is required, make sure the development environment is simple and straight-forward, and does not require years of specific or vertical experience.
- **Simple and powerful report writer:** Make sure the enterprise software or database used has a simple and powerful report writer. I know—the combination of simple and powerful is an oxy-moron in the software industry. But try to get as close as you can. My definition of a simple and powerful report writer is one that does not require a programmer, does not have a meta-layer and should take a week or less to

be productive. Custom reports are a must—if you get a good report writer, you will reduce the customization costs and bring down the hidden costs of an implementation.

- **Programmable interfaces:** Often new functionality requires an entirely new application to enter and maintain data within an enterprise software product. To add this new application, an Application Programmable Interface (API) must be available to validate and store data into existing tables. For example, an entirely unique inventory transaction might be required. An API should be available for all primary objects. An API usually involves an interface table, a process and a form to understand and fix any data transfer errors. APIs are a must for building interfaces between enterprise applications and custom add-on applications.

There are other flexibility features to look for in an enterprise software product, but these are rare and elusive. Therefore, if you can get any of the flexible features mentioned, you may be able to significantly reduce the customization hidden costs.

CONCLUSION

The primary intent of this document is to encourage companies to participate in the design and direction of enterprise software by demanding flexible functionality.

Enterprise software suppliers will take notice if more companies require flexibility. I recommend focusing carefully on flexibility when selecting an enterprise software product. Flexibility is the best way to reduce the large hidden cost of customizations. Users must take charge and steer the enterprise software industry, instead of letting software suppliers steer companies. Over the past 30 years, enterprise software has become more rigid, monolithic, complicated, and expensive. Keep in mind, the expense is not just in the cost of purchasing an enterprise software product. Instead, the expense also includes the cost of purchasing, implementing and managing the product. The solution should be flexibility—not standardization and compliance. It will be much easier to standardize on flexible software, than to make companies standardize on pre-defined business processes.

Casey Lawrence



About the author:

Casey Lawrence has over 25 years of experience marketing and implementing 10 different enterprise software applications to over 100 companies worldwide. Casey has held various technical, marketing, and senior management positions.