

WHITE PAPER
The hidden costs behind ERP software
Casey Lawrence, Enterprise Software Specialist
ABSTRACT

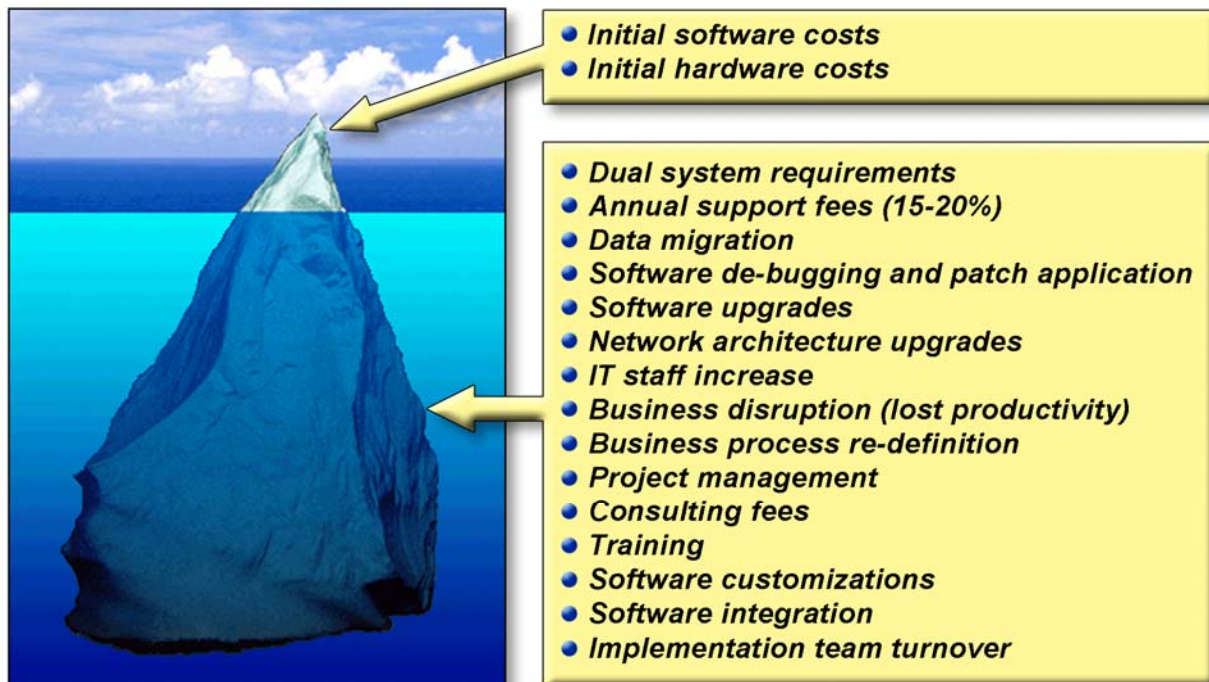
Discussed are the hidden costs behind enterprise software, including implementation, support and management. Many of these hidden costs become surprises along the implementation path. Therefore, before purchasing an enterprise software product, make sure you have a good handle on the hidden costs.

INTRODUCTION

I am sure we can all relate to a purchasing decision which later turned into a money pit, such as purchasing a car or house. During my youth I would often go boating with a friend whose father owned a boat. Often the boat would break down—usually just as we got to the middle of the lake. We would spend hours drifting in the water while his father would try desperately to resolve the problem. Eventually we would be towed to the marina where the battery, an electrical part or some other mechanical part would be replaced or repaired. One day his father told us what the word “boat” represents: “Bring Out Another Thousand [dollars].” I am sure boats are

built better these days, but it left a lasting impression on me. This experience has kept me from purchasing a boat, due to the potential hidden costs behind boating. Later, I learned firsthand about hidden costs when a reverse osmosis water purification system was installed under our kitchen sink. After the installation, I learned custom filters needed to be replaced by an expert repairman a couple times a year, translating into several hundred dollars a year, forever.

Unfortunately, we live in an age of supplier hooks. That is, suppliers would like to hook you on their products and services. From the beginning, the computer industry has been rife with these hooks. At first, it was the hardware companies, now it’s the software companies. Computer hardware has become a commodity these days and no longer carries the painful hooks of the past. On the other hand, software is a long way off from being a commodity, so plan on big hooks and nose rings to software suppliers. This is especially true with enterprise software, which includes ERP, SCM, CRM, WMS, PLM, etc. for all kinds of industries. Enterprise software is the backbone of most businesses.


Figure 1: The Hidden Cost of Software

Enterprise software is not only expensive to purchase, but comes with significant hidden costs.

Before purchasing an enterprise software product, make sure you have a handle on the hidden costs. The initial hardware and software cost is usually small compared to the hidden costs. I have seen hidden costs balloon over eight times the initial cost of the software. There is no escaping, all enterprise software has these hooks and hidden costs, including small products. Figure 1 illustrates the old iceberg theory (or should I say “law”?); namely, most software costs are hidden below the surface. The initial software and hardware costs are just the tip of the iceberg. Figure 1 lists most of the hidden costs required to implement and manage an enterprise software product.

Selecting an enterprise software product with low hidden costs will significantly reduce your total cost of ownership.

Nearly all software suppliers will claim their product has low hidden costs. Some of these suppliers may even have references to support this claim. However, don't be fooled, ignore the marketing hype, and make sure you get a good grip on the real hidden costs yourself before making a purchasing decision.

HIDDEN COSTS BEHIND SOFTWARE

The following list defines the hidden costs listed in Figure 1, along with some possible suggestions and pointers to help in the reduction or elimination of these costs:

Dual system requirements

Make sure you budget in enough hardware and IT support to manage dual copies of the software product and database on multiple systems. This means you may have one or more copies of the entire software and database on totally separate systems. The reason for dual systems is primarily due to implementation, upgrades and support requirements. You will need to build and maintain at least two complete copies of the software product and possibly more. One copy will be for your production environment, and another copy for training, testing, development, upgrades and debugging. (Usually this requires three copies.) Also supporting these dual systems could be significant. And, if you try to avoid these costs, you will pay in lost or poor productivity during and after the implementation.

Annual Support Fees

Most software requires support fees. These are referred to as software service agreements. Most software is considered a product you buy but not own. Unless you have developed the software yourself, you must pay to

use the software and/or be supported by the software supplier. Covered in most of these agreements is the use of the software, access to support and future upgrades. The cost of this can vary between 15% and 20% of the initial cost of the product. However, there are variations defining the cost calculation and it is different for every supplier. Make sure you ask some hard questions in this area up-front or you could get duped. Also, keep in mind, support fees are difficult to reduce after implementation or years of usage. The initial cost of software might go down over time, but your initial cost is fixed in stone, and it will be difficult to re-negotiate another support or usage agreement.

Data Migration

Migrating data from your old enterprise software product to a new enterprise software product is one of the most overlooked and expensive hidden costs within any implementation. True, data migration is a one-time process, but it requires weeks and possibly months of manually mapping data from old tables and columns to new tables and columns—one column at a time. Also, data migration is rarely clean. Usually, it turns into a convoluted mess, where multiple columns might merge into one column or a column might split into multiple columns, pushing the mapping effort down to the record level instead of the table level. And, this is not all; you will most likely have to scrub the data as well. Data scrubbing is the process of cleaning up your data before migration, such as deleting obsolete, unnecessary or damaged data. Don't take data migration lightly. I have seen projects painfully extended when users override management and demand data conversion instead of re-keying the data. Make sure all users and management agree to the data migration plan before beginning an implementation.

Software De-bugging and Patching

The process of de-bugging and patching has become another expensive hidden cost within enterprise software these days, especially for large monolithic products. There could be a significant number of patches applied during the implementation process as well as after the implementation. These patches will resolve problems and bugs with the software. The underlying code used when developing software is very inter-related, and in the age of monolithic software has become difficult to stabilize. Because of the inter-related development, software suppliers may enhance or fix one area of the software, and damage or impact other areas of the software, which in turn requires more patches. I am glad the word “bug” was used early in software history, because bugs reproduce rapidly, and this seems to be true with software.

The best way to work through bugs and get a handle on stability is to define all of your business processes up-

front, and figure out how to use the software for each business process. Test each business process and then fix the broken processes. Keep doing this until stability occurs. Then freeze the software from any more patches. This requires a lot of technical skills, which can be very expensive.

It requires an expert to identify a patch requirement, and, in most cases, another expert to apply the patch. Applying patches after you go live is even costlier, since you must apply and test all related business processes on a separate copy of the software product. Also, you should be aware, getting a patch is easy, but finding and qualifying the bug, and getting the supplier to fix the problem is an entirely different effort and cost. You can spend hours jumping through the hoops required by software support in order to get to the heart of the problem.

The hidden labor cost to identify a bug, get a patch, and apply the patch averages around \$1,000 to \$3,000 per patch. With the potential of hundreds of patches, this may be a significant hidden cost. I would love to have enough clout some day to tell a software supplier, "We will reduce the annual support payments by \$2,000 for each patch applied, since it is your software we are fixing." Remember, the amount of patches may be directly proportional to the size of the product.

Software Upgrades

Applying software upgrades has become rather painful and expensive these days due to the size and complexity of software. Performing an upgrade is not just running a process. It is now a major disruption to business, which requires making a copy of your production software and database, loading the upgrade (or upgrades) to the copy, testing all business activities on the copy, applying patches to the copy not included in the upgrade or broken during the upgrade, and doing this all over again in your production environment.

The cost to perform a typical upgrade for a medium to large software product will vary between 500 to 3000 man-hours depending on the size and complexity of the software product. And this will happen at least once a year, that is, if you want to stay current on the latest version. Therefore, it's best to get a good handle on the upgrade cost before making any purchasing decision.

You can reduce the upgrade hidden cost by upgrading long after the new version is released. This way, the version will stabilize after it has had a chance to mature and avoid the costs of being on the "bleeding edge." I suggest upgrading 6 to 12 months after the new version is released, unless, of course, you require the new functionality within the new version.

Network Architecture Upgrades

Several years ago, I was called to manage a large enterprise software implementation. After installation, we noticed the application was running slow. It turned out the company needed to replace their entire network architecture, including the replacement of all network hubs and lines throughout all of the buildings. This turned out to be a significant hidden cost and a major delay to the project. Make sure your network infrastructure will handle the new load.

IT Staff Increase

The decision to purchase and implement a new enterprise software product will most likely require new technical expertise. If the product requires a new database, you will need a Database Administrator (DBA) with expertise on the enterprise software product, called an "Application DBA." A good Application DBA cannot be created overnight and most likely will require someone with years of experience. You must either hire or contract this expertise, at least until your own IT staff comes up to speed. This will also be true with developers, that is, if you are planning on customizing the product.

Business Disruption (lost productivity)

The hidden cost of business disruption and lost productivity during an enterprise software implementation could be expensive and very difficult to define. There are three types of business disruption costs. They are:

- **Direct employee commitments:** A good implementation plan should define employee time commitments up-front (not including training). This direct labor must be paid through overtime or by lost productivity. This hidden cost cannot be avoided, but could be minimized if a good implementation strategy and plan is defined, which is beyond the scope of this document.
- **The learning curve:** Getting up to speed on a new enterprise product carries a certain amount of lost productivity. Employees will struggle to learn the new product causing a reduction in performance. This comes out in overtime and lost productivity. The hidden costs related to lost productivity can be rather obscure, hidden and expensive. There are two ways to reduce this hidden cost. First, develop, document and train users on the new business processes. Second, select and implement software which is flexible, intuitive, relational and easy to learn, which is beyond the scope of this document.

- **New or expanded business processes:** Often, when a new enterprise software product is implemented, new business processes are added and/or existing business processes are expanded. For example, you might implement MRP, requiring a new master scheduler, or the software requires several new steps to plan and perform a shipment, requiring more time or employees in shipping. Make sure these are identified and added to the hidden costs.

Business Process Re-definition

Integrating a new enterprise software product into any business will be time consuming. Someone must identify, understand, redefine and test all business processes impacted by the new software, along with any new business processes. This could be hundreds of business processes and could consume a significant amount of time for one or more individuals. You should be warned, trying to avoid this cost will just “pass the buck” and escalate the costs to the “Business Disruption” and “Consulting Fees” hidden costs.

Project Management

A good implementation project manager will be expensive. Do not try to cut corners on this position. A good project manager will be central to the success of any enterprise software implementation. The hidden cost of the entire implementation will be directly proportional to the availability, commitment and expertise of the project manager.

Consulting Fees

If you choose not have to have your own employees trained and used for implementation, you must bring in outside consultants with experience on implementing the software. The number of consultants will be directly proportional to size of the enterprise software and the length of the implementation plan.

Training

Training involves product, database and technology (if you are making enhancements or modifications) training. Off-site training averages around \$800 per day per employee, excluding travel expenses. If you have outside consultants, you may have them perform some of the training in-house. Make sure the training costs are reasonably well-defined.

Software Customizations

There are two types of software customizations: First, the addition of new functionality like new forms, processes or reports; and second, changes to existing forms, processes or reports.

If you are planning on adding new functionality, make sure you identify as many possible gaps prior to the selection and purchase. Don't be in such a hurry to purchase the software until customization costs are defined. Customizations often become the largest hidden cost during most implementations. A typical implementation begins with a small pre-defined list of customizations and ends up with significant unexpected customizations identified during the implementation process. These customizations skyrocket out of control, until a line is drawn between customization completed before and after implementation. Also, make sure you understand the development environment. Complicated development environments require highly skilled developers. The cost of these developers will significantly increase hidden costs. To understand the cost of customizations, have the supplier demonstrate a simple ad hoc customization process on-the-fly.

Customizing existing forms, processes or reports is messy and could threaten your software support agreement. Unless it is absolutely necessary, avoid the alteration of existing software. Most enterprise software is really rigid and does not lend itself well to modifications. If you must customize existing software, try to perform customizations with the least amount of impact to future upgrades, because customizations will significantly increase your upgrade efforts and costs.

Software Integration

If additional software products must be integrated to the enterprise software, the integration cost must also be considered. This might include bar coding, payroll, labor collection, testing equipment, EDI, report writers, business intelligence, etc. Products such as business intelligence might require the definition of meta-layers, which translates into weeks of setup, delaying implementation plans.

Implementation Team Turnover

Changing the project manager, consultants and users will significantly impact the hidden cost of an implementation. It is important to keep momentum. When there are staffing changes, the project must be stopped, redefined and re-started, losing critical momentum. Make very sure you have the right project manager, consultants and users committed before starting an implementation. Also, make sure all of these participants are committed throughout deployment, or you will spend a lot on re-training and re-orienting changes to project team members.

CONCLUSION

There you have it—my best shot at the hidden costs of enterprise software. Hopefully, I have provided some

enlightenment on the subject. As you can see, these costs are nothing to guffaw about. They need to be understood, estimated and taken seriously prior to the purchase of any enterprise software product. Nailing down all of these costs will be difficult and sometimes nearly impossible prior to procurement. However, making the effort to identify and estimate as many of these hidden costs will be well worth the exercise. Besides, no one likes iceberg surprises or hidden supplier hooks after a purchasing decision.

Casey Lawrence



About the author:

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