



BROWN SMITH WALLACE

CONSULTING
GROUP

Brown, Smith Wallace Consulting White Paper

Maintenance, Support, and Training

The Tripod That Protects Your Software Investment

September 2010

BROWN
SMITH
WALLACE

A graphic element consisting of a solid black square with three horizontal white lines of varying lengths extending from its top edge to the right.

A MEASURABLE DIFFERENCE™



INTRODUCTION

There is a great deal of discussion these days about the cost and value of maintenance, support, and training. This paper will examine real-world pros and cons associated with each area. It covers the theory and the reality of what is in the marketplace, looks at the economics, and makes recommendations.

DEFINITIONS

It is first necessary to have common definitions for each of the three areas to be covered. Some specifics may vary with different software vendors, but the basics are all the same. It is important to understand how this tripod of services supports the initial and continued successful usage of any software application.

MAINTENANCE

“Maintenance” is the practice of keeping a software package up to date and operational, meaning that the end users are able to properly utilize the various functions and features in their day-to-day business. It means the application is capable of successfully completing transactions in a manner expected by the user or described in the training and user manuals.

“Up to date” describes software following the actions taken that are required to keep the application current with best practices, government regulation, and industry specific requirements. It does not matter if it involves a new tax table for sales or payroll taxes, or the development of a report to satisfy a new government regulation.

Within the “up-to-date” category, there are multiple types of software updates. The most common are often referred to as “bug fixes.” These frequent updates repair problems found during normal operation.

A second type of update is the less frequent “enhancements.” These upgrades to the software represent improved functionality. They are often in response to end-user requests, as well as responding to changes in industry processes.

Finally, there are “new releases” or “modules” that provide significant improvements to the packaged software. In some cases, there may be extra costs associated with such updates as they add significant new functionality and are not just a simple enhancement over what is already available in the application.

SUPPORT

Support is the function of assisting the end users in their initial and ongoing use of the package. Support is often provided by software publishers, their Value-Added Resellers, and other third parties.

Support services often start with the installation and implementation of the software. Resources and expertise may be required for tasks such as conversion of older files, cleansing data that is missing or wrong, and setting of parameters that allow the software to function properly.

Once the package is operational, there is ongoing support. It may be to answer questions when things do not work as expected, to report and research potential bugs or problems, and to maintain interfaces with third-party software.

There are many different levels of support available. Some may be time-zone dependent (times of availability to match the times the business is operational) or controlled by Service Level Agreements (SLAs). Support may also be purchased for disaster recovery (actually an insurance policy to protect the business in case of a disaster or emergency). In the case of Software as a Service (SaaS) where the user does not manage the physical hardware environment, support includes backup and recovery, upgrades to the software and hardware as required, and other functions normally provided by an in-house staff.

TRAINING

Training is the process of teaching the end users and administrators how to use the package most effectively. Training starts during the implementation phase and continues when new employees are hired or when new functions are implemented. Ongoing training can be a great way to maximize usage of the applications by learning about new or unused capabilities on a regular basis.



Many software organizations and their end users also sponsor “User Groups” that meet on a periodic basis. These meetings are a great opportunity to obtain training, not only from the vendor/publisher, but also from actual users who have figured out different ways to use various sections of the software.

Training may be in person—either onsite or at a training facility—or remotely by utilizing web-based training tools. Self-study courses have the advantage of being available at anytime, anywhere.

THE PLAYERS

There are a number of different groups that are important to the discussion of Maintenance, Support, and Training. The first is the “software publisher.” This is the firm that designed, wrote, tested, and sometimes sells the applications. The publisher is ultimately responsible for the reputation of the package and overall satisfaction of the end users.

At the other end of the sales channel are the “end users.” These are the individuals and companies that purchase the software and use it to operate their day-to-day business. End users may be very sophisticated or simple. The companies may or may not have on-staff technical expertise to assist with the maintenance, support, and training related to the use of the software in any given environment.

In between, there are a number of other parties that may function in one or more roles. There are the Value-Added Resellers (VARs), who represent a sales channel that is commonly used by many software publishers. VARs have different levels of expertise and capabilities. Some are only involved in the sales activities while others are active throughout the whole software lifecycle.

Independent Software Vendors (ISVs) provide technical capabilities to expand those available from the publisher. Frequently, their expertise is used to develop and integrate specialized functionality for vertical markets. An ISV may provide all three functions (maintenance, support, and training)

for their “add on” capabilities. An ISV is usually “approved” by the vendor, and its products are certified to meet the quality control requirements of the software publisher.

There are also third parties that participate in software projects. Some provide additional software, but it is not always certified by the publisher and often requires technical expertise to interface it with the application.

Third parties also have built businesses around offering support to users of certain publisher’s software. These companies usually grow out of end-user communities. They are able to augment the publisher’s own staff on a timely basis.

EXPLORING THE TRIPOD

From an historical perspective, there are changes taking place in the marketplace that are modifying the way end users evaluate the cost of services and make decisions on expenditures.

Ten years ago, many end-user companies made a conscious decision to eliminate maintenance and support after three or four years. Today, most companies continue their ongoing maintenance and support contracts.

The change in attitude appears to be due to a transformation in strategy related to software purchases. In the old days, it used to be that software was replaced relatively frequently (every eight to ten years). That meant that once an application was up and running, the users could decide to just continue to use it “as is” until it was ready to be replaced. At the then-current annual cost (at that time it averaged about 18% per year) the savings from year four through year eight would represent a dollar figure close to the purchase price of the next package.

Today, more companies are interested in extending the useful life of the software investment. Since conversion and training are the most expensive elements in implementing new software and the ability to stay up to date is much less expensive than learning a completely new application, many executives now consider their maintenance budget as prepaying for the next software package.



In addition, as the software industry has matured, the publishers have learned to continuously improve and upgrade their offerings. The old-fashioned reasoning to change applications due to obsolescence is almost nonexistent.

If maintenance, support, and training are part of a strategic plan to extend the life and value of a software package, how can they best be used through the lifecycle of the application? Part of the answer is driven by understanding who offers the various services.

The publishers are the first line of investment protection. They maintain the integrity of the package. It is their research and development staff that is constantly looking for new and better ways to accomplish every business purpose. They get input from industry gurus, end users, and prospects as to wants and needs in the software package.

All of the input is organized, analyzed, and sometimes sent to advisory councils to help with the prioritization and scheduling of enhancements to the package. The publisher then publishes a list of new capabilities that are to be programmed and provided in the next couple of releases.

This advance notice allows end users to plan for new functionality, consider ways to improve their usage of the software, and generally feel positive about the direction the publisher is taking.

In those few instances where a real problem (as opposed to a misunderstanding of how the software is supposed to be used) is found in the application code, it is the publisher's programming team who will diagnose the situation, plan and implement a fix, and then deploy it to all users with instructions. Usually, there will be advance notices as to what was found, short-term work-arounds, and expected timing for release of the fix.

These experts are worth their weight in gold when something goes wrong. They are well versed in the inner workings of the packages and can track down and fix most issues in a matter of hours, if not minutes. They develop "knowledge bases" of what others have already been through and often have a solution immediately.

This group is also on the alert for common issues that may represent a new way of doing things. When customer support sees a pattern of calls, they can alert the R&D staff that something needs to be made simpler, faster, modified in some way, or generate a "training bulletin" to reduce or eliminate the calls related to that issue.

When the sale is made through a VAR, the staff of the VAR is frequently the first point of contact for questions or concerns. This is an efficient way to funnel problems "up stream." The VAR will be given access to the knowledge base, have frequent conference calls with other VARs, and be able to respond quickly and efficiently to the common questions from the end users.

When an issue is found that is not easily answered, the VAR is in the best position to do the initial diagnosis and provide a well researched report to the publisher so its team can begin its analysis with good descriptions of the problem and basic research completed.

Where third-party packages are integrated with the base application, the VAR is also the control point to keep all of the correct players in the game. The VAR can coordinate activities and assist with continued diagnostics onsite as necessary.

When a package is first acquired, the support functions kick in—in high gear. The successful use of an application is dependent on getting a good start in its use. Implementation support services are almost always an extra charge. They should be proposed as part of the initial purchase. This level of support is often provided by the selling organization.

Many publishers use VARs and other resellers as their sales and delivery channel to the end-user market. This is very effective, especially when the reseller is well versed in the specific vertical market of the buyer. Domain expertise almost always ensures a smoother transition to a new application. The support team talks the language of the end users, and this always makes it easier to be successful. There is less misunderstanding due to terminology associated with the business.



Installation and implementation support often fit into the following general categories:

- Installation (loading the software on the local infrastructure and verifying that all users have access to the applications)
- Conversion (converting data from an existing system—manual or automated—to the new automation)
- Setup (understanding how the application is to be used and setting parameters and “switches” that personalize the way the application is used)
- Capability Testing (validating the operation of all subsystems)
- System Testing (validating the proper operation of the complete application; processing test transactions and exercising all aspects of the new software)
- Go Live Support (“handholding” on the first day, days, or even week of initial use)
- Post Startup Audit (a review of actual use of the application compared to expectations; this usually includes an ongoing plan to continuously improve and expand the use of the applications)

Post-implementation support is usually focused on improving usage of the applications. The first level was described above in the discussion of first responder support in case of problems or issues in using the application.

Additional support can be provided in a number of consulting processes, including:

- Process improvement.
- Long-term information planning.
- Infrastructure improvement.
- Strategic planning (the most successful system implementations directly support the strategic plan of the organization).
- General innovation (finding new ways to do things).

Be careful not to treat support as training. Training is used to make sure people know how to properly use the application.

Support calls for basic training creates potential problems for companies needing support to resolve critical issues. It ties up resources and may create backlogs of requests for help.

When a company abuses the support line, it may get higher charges or be asked to revise how users call in for support. A publisher or VAR may require a single point of contact from the end user. That person would be trained to access the knowledge base and answer the majority of questions from poorly trained users.

Support can also play a role in customization and working with onsite technical personnel to better use the applications. There are even companies that provide a full range of consulting service to improve use of the application system. These services include reengineering of workflow, application of best practices, and often strategic planning support. It is important to recognize that the publisher cannot be responsible for fixing problems caused by “add on” code that is not of its design and creation.

The last leg of the tripod is training. One of the most short-sighted things many businesses do is during final negotiations for a system is decide that their people are smarter than average and do not need all of the training being proposed by the vendor. In most cases, more training, not less, is required.

Training may be offered by any of the members of the players defined above from the publisher down, stopping at the end user. Publishers normally provide the more robust remote training. They have the investment dollars to develop interactive programs that can be viewed on the web from anywhere at any time. This is a great advantage when large numbers of people need training with vastly different schedules.

One great idea for training is to develop in-house experts. These “go to” people learn sections of the application in great depth. They spend the time to get into how the application works, what it can do for the organization, and experiment to find better ways to use the purchased capabilities.

Having your own experts is a value in another way. It encourages the end users to take more control of their own workflows. The IT group is no longer some outsider telling you how to run your business, but a bunch of friends and coworkers who know and understand what you are doing, why you do it



that way, and the ramifications of changing what works. The insider has many fewer problems getting acceptance of new and better procedures.

Training during the start-up phase is very important, but it does not stop there. At first, it is important to learn how to do the basics and how to do them well. As the routine operations become smoother, it is time to get advanced training on new functions and different ways to accomplish many goals. Advanced training has the potential to increase the return from any software investment many-fold.

Ongoing training is also valuable when turnover occurs. In most companies, the experienced users train the new employee in what has to be accomplished to get the job done. There is little thought given to understanding the package or its capabilities. Having outside training increases the chance that new ways will be discovered and then used to make the process more effective and increase the return on investment for the software.

Another training opportunity is attendance at the "user group" meetings. These training occasions are special as groups of users from many different companies get together. It is a unique opportunity to gain insights from other companies, learn secrets that are not printed in any manual, and find ways others have tweaked the application to get added value.

A special set of meetings is often set aside for the top executives. These are also very valuable and give executive management a new view as to the possibilities. "C"-level executives and company owners come back from these meetings energized and with a better understanding of what is possible.

How do you utilize maintenance if you have customized the application? Sometimes you have to reprogram all of the changes to utilize an enhancement. The first choice is not to customize the code. Use Application Program Interfaces so that all updates can be applied without disturbing the "add on" application.

COSTS

Any discussion of maintenance, support and training always ends up with a discussion about cost. What does it cost, is it justified, and can I find ways to reduce my expenses?

The first question is: What does it cost? There is no single answer. Industry average ranges from a low of about 18% (figured on the retail cost of the software purchase) to 25%. The range is somewhat dependent on the delivery system of the publisher. Where VARs and other third parties are involved, there is usually a higher cost.

The higher cost is justified by the availability of local support services that can be very personalized. Where customization occurred, having access to the programmers in an ongoing relationship is very worthwhile. They know what was done, how it was done, and what effects the changes should have had. They are the best situated to be able to diagnose and correct any problems.

Maintenance has a secondary value to the end user that is usually invisible. That is, it provides a continuing source of R&D funds to the publisher. Software companies know that their users depend on them to keep the value of the software high. The best way to do this is by keeping pace with the industry and user requirements. If publishers allow their packages to fall behind, not only will new sales dry up, but the embedded base will erode quickly.

These fees can be viewed as an investment in the publisher to keep increasing the value of the package. It is an unwritten contract that works very well.

Specialized support is almost always an extra cost service. These are provided on an as-requested basis. Not every sales organization has the in-house capability, so it is important to ask in advance if that is of importance in the selection of a software vendor.



SERVICE-LEVEL AGREEMENTS

In any contractual situation, it is possible to change the parameters of standard service. There will be a cost, but it may be worthwhile. The Service-Level Agreement (SLA) provides a framework for services required by any end user.

A few of the issues normally covered relate to time, responsiveness, and minimum service delivery. Time is usually related to time zones. With many software companies housed on either coast, there may be gaps in service for those on the other side of the country. For example, a New York-based company with support hours from 8 a.m. to 5 p.m. would be closed at 2:00 in California. That is right in the middle of the business day.

Adding service until 5:00 Pacific Time is not difficult, but there will be a cost. At the time of initial negotiations, request quotes for the extra service necessary to cover your business hours. Do not forget weekends if that is a busy time for you. Most companies are not available on weekends without significant charges.

Responsiveness is another area where contractual SLAs are helpful. Most publishers have standards already in place. The common agreements separate calls into categories of criticalness. (One cannot use “importance”—everyone’s issues are important.) There are at least three levels that will be encountered.

1. Critical—the system cannot be used
2. High—the operation has found a way to work around the situation, but it needs to be corrected quickly
3. Low—the problem is not affecting production, but it is an irritant or is interfering with efficient operation

The response times will be related to the level of urgency. A type-one (Critical) issue would have a response time of less than one hour (during normal business hours) from the time of the call to a first response. It would normally carry with it a guarantee that the problem will receive high priority until it is resolved.

Where a VAR is involved, the SLA may require an onsite visit. This is an advantage to local support—especially if the users do not have a sophisticated IT department to assist with problem resolution.

Type-two (High) situations usually require same-day (if initiated by 2:00) response. Again, a local VAR can probably provide quicker response, is more familiar with the specific operating environment, and has a significant incentive to maintain a positive, ongoing relationship with the customer.

Type-three (Low) issues are usually set for the next business day as a first response. In all cases, the publishers and VARs are interested in securing a reputation for fast support as that becomes a competitive advantage in making new sales. They are there to help the end users be successful and take the responsibility seriously.

Many publishers have developed self service application capabilities to start a “support ticket.” These database programs often use artificial intelligence to walk the user through a set of simple questions to focus on the problem. In many cases, answers are developed during this analysis that resolves the situation.

More of these applications will be developed in the future. The highest cost in any support environment is people. If automation can be used to better leverage these resources, everyone wins. The trick will be to design the interface so that the end user feels cared for and not just converted to a data input machine.

IMPACTING THE FUTURE

Every publisher that is in the field for the “long haul” is constantly looking for ways to improve its package. Publishers want to find competitive advantages they can use to sell more software. One of the best places to get the information is from current users.

If you have ideas—or needs—make sure to understand the internal processes of the publisher for accepting, analyzing, and selecting enhancements to go on the development calendar.



BROWN SMITH WALLACE

CONSULTING GROUP

Working within the system can be very beneficial. Not only do your ideas get heard, but you may be given other ways to accomplish the same end without having to wait for new enhancements to be developed.

In many cases, you will find the developers are already working on something very close. With a small amount of “tweaking,” what is already in the works can be delivered with your specific needs covered. The more visible (from a positive sense) you are to the R&D people, the more chance you will be asked for input and suggestions when ideas you might be able to use are proposed.

There is only one potential downside, and that is your idea and any advantage it would produce for your company will be shared with the rest of the users. In cases where an upgrade would provide significant competitive advantage, it may be worthwhile to have a custom modification developed with the clear understanding that it will not be shared with others without your specific written permission.

In cases where there is a sudden regulatory change or other operational requirement that affects the whole industry, you can usually rely on the fact that the publisher is tracking the potential changes at all times. Publishers normally prepare high-priority changes to the software to respond to any such situation. Still, it would not hurt to drop the support group a note alerting them to a future change as soon as you hear about it.

CONCLUSION

Maintenance, support and training are keys to a successful ERP implementation and ongoing use. Packaged software is a complicated product with millions of lines of code, hundreds of interfaces, and thousands of users who are always trying something new.

Keeping everything working in this environment is an investment on the part of all members of the team; from the publisher through the channel partners to the end users. Given the maturity of the industry, the cost for maintaining the applications can be realistically considered as a better alternative to purchasing a new package every eight to ten years. It is much less disruptive on the operation and extends the value of the training time for many more years.

Incremental improvements in operations, along with support services will keep the software investment viable for a much longer time. Staying involved with the publisher will allow any sophisticated users to affect the future direction of the application and almost eliminate the need to change software packages on any scheduled basis.

Together, maintenance, support, and training will add life and capability to your software investment. The return on investment will be increased and the long term value of the software will be assured.



BROWN SMITH WALLACE

CONSULTING GROUP

ABOUT THE AUTHOR

Steve Epner is the founder of the Brown Smith Wallace Consulting Group. Steve is also an adjunct professor of Innovation and Entrepreneurship in the graduate program at the John Cook School of Business at Saint Louis University and is a member of the Center for Supply Chain Management Studies. In addition, Steve is a member of the faculty at the University of Industrial Distribution and Arizona State University for the Certified Professional Manufacturers Representative program. Steve has a Bachelor of Science in Computer Science (1970) and a Master of Science (2005) from Purdue University.

ABOUT BROWN SMITH WALLACE CONSULTING

The Brown Smith Wallace Consulting Group has been serving the distribution community for more than 20 years through the publication of various Software Guides, an online evaluation center and resource center at www.software4distributors.com and assisting companies who need help selecting the best software packages for their business and maximizing the benefits from their investment.

BSW Consulting

10151 Corporate Square Drive
Suite 100
St. Louis, MO 63132

314-983-1200

www.bswllc.com

www.software4distributor.com