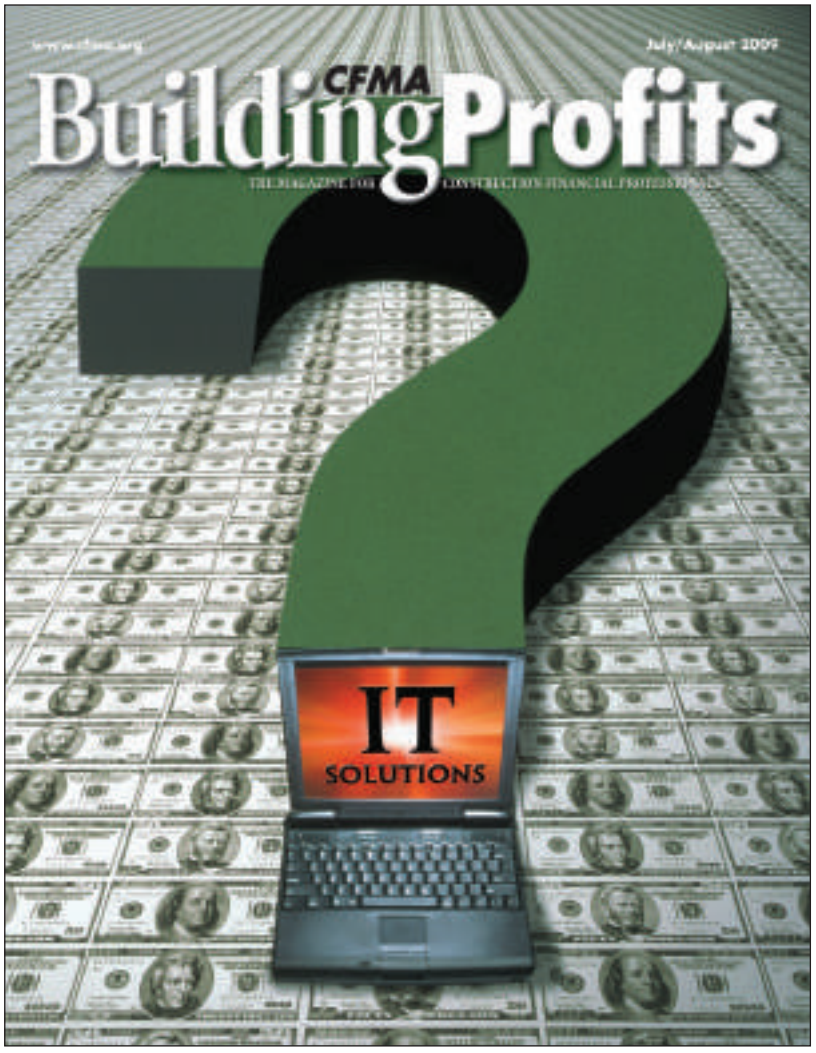
CFMA Building Profits
THE MAGAZINE FOR CONSTRUCTION FINANCIAL PROFESSIONALS

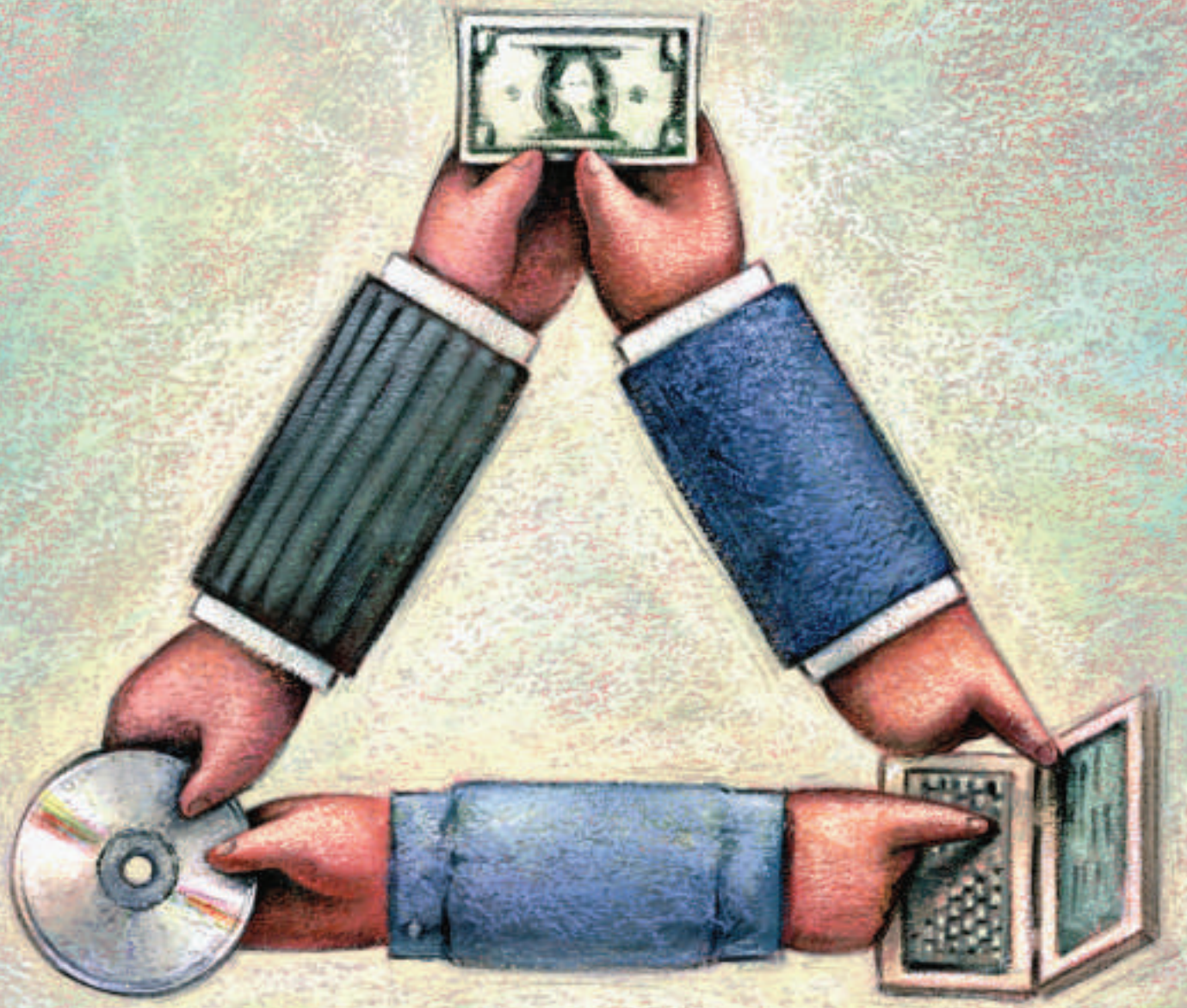
REPRINT



July-August 2009

CONSTRUCTION FINANCIAL MANAGEMENT ASSOCIATION
The Source & Resource for Construction Financial Professionals

SOFTWARE IMPLEMENTATION BEST PRACTICES



While 76% of respondents to a recent CIO Insight survey said, "Enterprise systems provide a competitive advantage and strategic value to their organizations," they also reported that (on average) their companies use only 64% of their enterprise systems' core functions.

BY LAURENCE C. TRUE & KURT M. KOENIG

In these trying times, software vendors and consultants like us are watching evaluation and implementation trends for new systems with keen interest – after all, it's our livelihood. Those of us who have been in the industry for decades have noted some meaningful patterns:

- Most construction companies that are big enough to have CFMA members also have "enterprise software." This software is used company-wide and integrates such primary functions as job costing, accounting, payroll, equipment management, etc.

  This is in contrast to departmental software (such as stand-alone project management or sales force automation systems) and personal productivity software (such as Microsoft Office).
- Many systems have been in place for years, resulting in operational practices and policies that have evolved to negate shortcomings in the software.
- The latest and best trends in enterprise software focus on integrating more and more business functions under the enterprise umbrella.

  For example, one key trend in construction enterprise software integrates project management and administrative functions with job cost and accounting, and adds workflow and document management capabilities to the enterprise suite.
- Even though older legacy systems lag behind these trends, most companies find the process of selecting and implementing a new enterprise system an easy decision to defer.
- Companies that are well-positioned to weather the present economic storm – stable, sound, and well-run enterprises – will take advantage of this slow period to re-evaluate and upgrade not only their software, but also their key processes and procedures, and will emerge as leaner, more responsive, and more effective competitors.
- Far too many software applications end up as "shelf-ware." After they are purchased and the money is spent, they never achieve their stated objectives.

Regardless of the economic climate, implementing new enterprise software is a rare event. Our observation is that the typical contractor changes systems every 10 years or so. Therefore, changing systems presents an unusual (and perhaps even unique) opportunity to really improve how your company operates.

Given the huge upside potential and significant impact, when your company finally decides to make a change, it's essential to get it right. Applying best practices to the implementation process can lead to the broader deployment of best practices in all key business processes, and significantly increase the overall benefit of making the change.

The purpose of this article is to provide a road map to successfully implement new enterprise software. One piece of good news for our project-driven industry is that every software implementation is a project. As such, applying standard project management disciplines will help ensure the success of your software implementation project.

What Is a Project?

A project, whether software implementation or construction, is a complex series of nonroutine tasks directed to meet a specific one-time goal. All projects have the following characteristics in common:

- *They are unique in nature.* While similarities between projects often exist, each project is different from the last. And, even though it's beneficial to have people on your team with software implementation experience, all implementations present unique challenges and opportunities.
- *They have a defined timeline.* Projects have clear start and end dates, and sometimes critical milestones. Deliverables must be produced to meet a specified set of objectives. Although changes in scope and resources occur, managing the schedule is critical to success.
- *They have an approved budget.* Projects have time and cost budgets – the deliverables must be produced to meet specific customer requirements.
- *They have limited resources.* An agreed-upon amount of labor, equipment, and materials is allocated at the start of a project.
- *They involve an element of risk.* Projects entail a level of uncertainty and, therefore, carry business risk.
- *They achieve beneficial change.* The typical purpose of a project is to produce a desirable end product – in this case, to improve your company through the implementation of business change.

Project Planning

In the immortal words of Yogi Berra, *"If you don't know where you are going, you might wind up someplace else."*

If you apply this wisdom to manage an enterprise software implementation project, beginning with a good plan greatly increases the likelihood that your company will arrive at its desired destination on time and within budget. Your company's implementation plan should address the following:

Scope, Goals & Objectives

During implementation planning, it pays big dividends to take a structured look at your goals and objectives – from both a personnel and functional perspective.

Start with the well-defined business outcomes you expect as a result of the process. You should state your objectives in business terms, without referring to computers or software. This provides a better framework for setting priorities and measuring project success than phrasing goals in terms of software features.

For example, instead of stating that, "We want an automated A/P invoice routing and approval application," state what *results* you expect from automating this process. Some possibilities include:

1) Reduce the A/P processing cycle time from two weeks to two days.

2) Process all invoices so that all available vendor discount terms are met.

3) Reduce the cost of invoice handling from $20 per transaction to $5.

4) Avoid paying vendors without securing appropriate documentation.

When stated in quantifiable business terms, these outcomes can serve as the benchmarks against which you will measure success. It's one thing to say, "We implemented A/P invoice routing." But it's more significant to say, "We achieved the following cost and time savings . . ."

Your business objectives also provide a framework to help determine what's important in the fray of implementation: "Yes, we are adding labor in the mailroom to scan all the incoming invoices. But, the downstream cost savings more than justify the expense."

Top Management Buy-In

Management must be committed to the cultural and procedural changes that are key to reaping the real benefits from changing enterprise systems. This point was emphasized repeatedly in the panel presentations on "Software Imple-

One piece of **GOOD NEWS** for our **PROJECT-DRIVEN** industry is that **EVERY SOFTWARE IMPLEMENTATION** is a **PROJECT.** As such, applying standard **PROJECT MANAGEMENT DISCIPLINES** will help ensure the **SUCCESS** of your software implementation **PROJECT.**

mentations: The Real Story" and "Use of Workflow & Technology to Improve Timing & Efficiency for Accounting & Project Management Documents" during CFMA's 2009 Annual Conference & Exhibition.

The Workflow panel handout specifically asked, "What is the (company) owners' level of commitment and enforcement to each?" In response, the panelist from L.P.R. Construction cited "Our management team bought in" as the top item on its list of "reasons for success."

Successful companies are rarely pure democracies – input from the team is valuable, but at some point, management must set the direction and follow through.

Gap Analysis

Review your company's current vs. intended processes and procedures to determine if there are any gaps between the two. The new enterprise software will almost certainly handle processes differently than your old software, and might not address all the functionality your old system did.

Some gaps, however, are not necessarily bad – they may just indicate process flow variations between software. Identifying these gaps at the front end of the process, as well as working them into the implementation plan, reduces the chance of disruption once the project is under way.

Change Management Procedures

Scope changes are as much a fact of life in a software implementation project as on a construction jobsite – and, it's just as important that you have defined processes for handling these changes in your project plan. Here are some examples of such scope changes:

1) Increase the footprint and complexity of the project by adding either business units or software modules that were not included in the original project plan.
2) Introduce software modifications that were not in the original plan, or changes to the scope of modifications that were in the original plan.
3) Request additional professional services (such as training, data conversion assistance, etc.) from the vendor or consultant.

Similar to a construction project, it's important to document the reason for the change, its impact on cost and schedule, who will be responsible for monitoring it, and who will pay for (or absorb) the cost.

In our experience, a formal process with documentation and sign-off for any change that will materially impact the cost or schedule is the best approach. *Undefined scope creep* has led many implementations off track and left CFMs to face tough questions after-the-fact.

Roles & Responsibilities

The quality of your implementation team and the amount of time made available for the team to do its work will materially impact the process and final result.

Due to the new integrated software, your team members will have to make many decisions about process changes, including potentially sensitive interactions between departments. (Specifically, people who rarely interact with each other may need to work together to resolve workflow issues.) Therefore, it's important to assign personnel who really understand your company and its operations.

One important qualification for all team members is a positive attitude – especially for those in leadership roles. Change can be difficult, and the "vibe" your users get from your team can set the tone for acceptance, which is essential to realizing your project goals and objectives.

In addition, there must be enough time to complete the implementation project – the best team in the world can't get the job done if it doesn't have enough time to do it properly. If necessary, hire temporary or new personnel. It may be helpful for them to pick up as many routine duties as possible, so that your team can focus on the project. Your enterprise software implementation team should include the following roles:

Executive Sponsor

This person is committed to the success of the project and must have the authority to make the tough decisions across a number of critical areas. He or she must be able to:

- Protect the team from distraction, reassignment, etc.
- Make the final call on scope and schedule changes.
- Adjudicate differences of opinion between functional areas when introducing new integrated software and business processes.
- Communicate the high-level "why and how" of the new system to other stakeholders and users in the company when the inevitable resistance to change occurs.
- Keep the "big picture" in sight and not get distracted by the details.

Implementation PM

The implementation PM works to balance and manage the project's scope, schedule, and resources. In that sense, project management skills and experience can be more valuable for this role than familiarity with every detail of your company's business processes.

Though not necessarily the ideal candidate, the CFO is often chosen as implementation PM. Instead, consider appointing a respected construction PM from your own operations; doing so will help bridge the communication gap between operations and accounting.

Champions for Each Functional Area

Many of the recent advances in construction enterprise software include better integration between (formerly) disconnected departments. The integration of accounting and project management with process workflow are perfect examples. Making the best use of these tools can involve major changes in how your company operates and how its departments interact. This includes:

- Standardization of processes that have traditionally been off-line.
- Coordination of information (such as customer and vendor records) that used to be handled separately by project management and accounting applications.
- Formalization and documentation of business processes that were previously handled according to company tradition (such as document storage and retrieval).

Your team members should be able to knowledgeably and authoritatively address the key functional areas of your company. Operations and accounting personnel will have to work together – a novel concept in many companies.

Your Vendor

Your enterprise software vendor shares your company's goal of a successful implementation. Thus, it's important to remember these points:

- If your vendor makes a recommendation, it's usually because it was successful with other clients.
- In the beginning, your vendor will know the system far better than you do, so it's important to defer to the expertise of its training and support staff.
- Since your vendor depends on your company having a well-trained and efficient staff in order to use the system properly, its training recommendations are well thought-out and should not be considered as "pushing for more services."
- Your vendor should be viewed as your partner in the implementation process – not an adversary. Remember, your vendor has your company's success as a primary goal.

Consultants

Consultants can help your team, but they should not be doing its work. The worst scenario: Your consultant walks out

the door with half of the software implementation knowledge because your team relied too heavily on him or her and did not take ownership of the system.

Budget

The key to establishing your implementation budget is to be realistic, so be sure to include internal and external costs. The following rules of thumb are typical for total cost breakdown on a well-run enterprise software implementation project:

- Hardware = 15-20%
- Software = 20-25%
- Implementation = 55-65%

All companies must spend their money wisely, so while cost cutting is tempting, doing so with hardware/implementation costs *after* spending thousands of dollars on software is not a sound decision.

Hardware and infrastructure (networks, etc.) must deliver acceptable speed and performance. End users are easily frustrated by slow software – no matter how easy it is to use or how many wonderful things it does. And far too often, enterprise systems don't achieve their intended goals because of inadequate hardware.

Buy-In

As you create your implementation plan, be sure to get buy-in and acceptance from key stakeholders. Natural resistance to change is minimized when people feel part of the process. However, that does not mean you need to seek consensus on every question.

In fact, your executive sponsor and the entire implementation team will be hard-pressed to make everyone happy on every decision. Here are some examples of differing, and possibly conflicting, perspectives:

1) Management may want more and/or better information.
2) Users may want to get their work done faster and easier.
3) More information to management might require changes in job duties for some users.

When everyone understands the trade-offs and their importance, and is confident their thoughts and concerns were taken into account, then acceptance of the outcome is much more likely.

Schedule

Stakeholder confidence in the successful outcome of the project is enhanced by setting and meeting target dates and deliverables. Establish a realistic schedule that includes specific tasks, dependencies, responsibilities, and high-level milestone dates.

Remember: Manage according to the schedule and keep it up-to-date with changes. Also, keep stakeholders informed about milestone dates and include the reason(s) for any change(s).

Metrics & Reporting

For the last part of the planning process, lay out the metrics you will use to manage the project and communicate them to all stakeholders. These metrics typically include summarized measurements of scope, schedule, cost, and personnel (as previously discussed). In addition, consider measurements of project quality and risk factors.

Communication, Communication, Communication

Implementing new enterprise software can affect everyone in your company. And, the predominant concern will almost always be change. The following are some common implementation fears that must be addressed:

"I just got the hang of our existing system, and now I have to learn something new?"

"I can barely get my work done now, and I'm afraid the new system will make it harder!"

Given the **RATE OF GROWTH** in **HARDWARE CAPACITY** and **SOFTWARE CAPABILITY,** a good rule of thumb is: If it's **THREE YEARS OLD** or more, it's **OBSOLETE** or nearly there.

"Could this affect my job security?"

"I sat through a demonstration of the new software and it looks way too complicated. I'm not sure I'll be able to use it!"

The best way to address the fear, uncertainty, and doubt among your staff at the outset (and the potential for misinformation to spread "through the grapevine" during the project) is through effective communication at regular intervals during the project life cycle.

Project Kickoff

Here are some pointers to maximize the implementation of your new system:

- A *project charter* is a great way to summarize the result of your planning process – post it where everyone can see it.
- Hold a *kick-off meeting* for the implementation team. Review the project charter and details of scope, schedule, budget, roles and responsibilities, reporting, etc.
- Encourage departmental managers to hold *regular meetings* within their groups to hear and address questions, concerns, and suggestions.

During Implementation

Schedule and conduct regular status meetings. These meetings fall into two categories: strategic and tactical.

- *Strategic meetings* usually involve the entire team and serve to update everyone about project status, changes in scope or schedule, roadblocks, or issues that must be addressed.
- *Tactical meetings* are for certain team members to work out specific issues (for example, the details of payroll burden setup and its impact on job cost, accounting, and billing).

The team should also communicate regularly with all other stakeholders. Some of the most successful implementation projects we've worked on featured regular newsletters that provided high-level summaries of project status and other topics addressed in the strategic meetings.

Newsletters also present an opportunity to publicize and celebrate project successes (such as the completion of key milestones) and to recognize the efforts of team members who go "above and beyond" to keep the project on track.

Training, Training, Training

Training is almost always the implementation component with the highest payback. Well-trained users will:

- Use the software more effectively to realize your company's goals;
- Be happier because they will be spared the frustration of figuring out how to use the new software on their own; and
- Make fewer errors and, therefore, improve efficiency and enhance the likelihood your company will realize its business objectives for this project.

Remember: Understanding an integrated system with controls and audit trails that affects your entire company is more complicated than learning how to create Excel spreadsheets.

Effective enterprise software training goes beyond helping users understand how to use the software – it helps put the applications, processes, and procedures into the context of the business outcomes your company wants to accomplish.

When users understand why they are being asked to provide more information or work in a more structured environment, they are much more likely to accept and work with the new system.

The best practice is to design your company's training program to address the needs of two classes of users: 1) your team's functional experts and 2) your company's end users.

Team Functional Experts

This group will make decisions about business practices and software configuration. Therefore, these team members will need more comprehensive training to understand all of the software's capabilities and to make the right decisions about how your company will use them.

Company End Users

As the name implies, this group will use software as it is configured. End-user training must be more concise and specific – literally, "Here's how you do it." *And, the trainer needs to understand what these users need to know before training begins.*

Presenting end users with a dazzling array of system configuration options leads to confusion. End users need to understand the goals that were set and how their work will change. It's best to pare the training down to the essentials. This also reduces training time and cost. The following examples should prove helpful:

1) To ensure that our company is paid for all work performed, we must have signed approval before proceeding with any work outside the original project scope.

Therefore, management needs visibility of all potential, pending, and approved change orders, including description, status, and any costs incurred to-date.

Therefore, PMs will enter all potential change orders into our new enterprise system, maintain status information, and instruct field personnel to segregate change order costs in reporting.

2) A/P invoices must be booked and available for management's review within two days of receipt. Field managers are responsible for approval of open invoices no later than three days prior to the discount date. When a discount is lost due to late approval by the PM, the lost discount will be charged to the project.

3) At the time material orders are placed, PMs must provide proper job cost coding so that invoice processing at the time of receipt is not delayed. (At this point, the documentation should describe the process in detail and how it integrates into the software.)

Hardware

Enterprise system performance is essential to user acceptance. A quick response leads to happy, productive users – and a slow response leads to frustrated, unproductive users. In addition to a quick response, appropriate computer monitors are an essential and often overlooked item.

For example, if you are implementing integrated document management and workflow, be sure to provide either widescreen monitors or dual monitor setups that allow users to view adequately-sized documents and work in the software at the same time. This is especially important for users who perform such document-intensive tasks as invoice entry.

Be sure to get your software vendor's input and recommendations to help ensure that all aspects of deployment are appropriately configured for anticipated usage levels. Items to consider include servers, workstations, and a network.

A network has many facets, including: 1) the internal network (the local area network within your office); 2) the remote site network (the type of connections and bandwidth your company put in place for remote offices and jobsites); and 3) all factors impacting network traffic and performance (such as transaction volumes, documents, graphics, and other applications).

Unless existing hardware meets or exceeds your software vendor's recommendations regarding speed and capacity, avoiding the purchase of new hardware can lead to substandard results. Given the rate of growth in hardware capacity and software capability, a good rule of thumb is: If it's three years old or more, it's obsolete or nearly there.

Testing, Testing, 1-2-3

Two clichés that apply to software implementation are: "The best surprise is no surprise," and "Never assume anything." Basically, when you're ready to roll out your new enterprise software, everyone will be happier if it all works as anticipated.

The best way to achieve this result, or get as close as possible, is to test all functions thoroughly. This process is sometimes referred to as a "conference room pilot" because it often entails the team sitting in a conference room and running through many scenarios, checking results across departments, and double-checking functions.

Here are some pointers for a successful conference room pilot:

- Use your company's real data;
- Evaluate results for accuracy by ensuring that the setup produces the desired results;
- Evaluate processes to see if they match how you want to do business;
- Adjust setups and configuration as needed; and
- Retest until comfortable.

The main thing to keep in mind when allocating time and resources to testing is that ***getting things right is much more cost-effective during implementation than after rolling out the new enterprise software to the entire company.***

And, you will learn a lot about how the software really works and what impact it will have on your company. This leads to the next topic – modifications.

Out-of-the-Box Can Be Easier Than You Think

One way to save both money and headaches is to defer or avoid making modifications to the software until after you have it fully configured, tested, and (ideally) in use for six months or more. Our experience is that most upfront software modifications are later changed or abandoned.

Why? Because it's virtually impossible to fully understand all the nuances of a modern, integrated application suite until it has been used for awhile. Once the software is fully understood, your perspective on needed information will shift. You will make better judgments about what changes you really need when you fully understand the software and how your company uses it.

Sometimes, people will insist on having their old reports reproduced in the new system because they think they'll need the familiar reports when they "go live." This is often a wasted effort because the new system should address those management needs differently and more effectively.

When you think about making modifications, carefully consider changing processes before you change the software. Conversely, if you are going to change the software, make sure you really understand your company's processes.

Our experience has shown that many changes are requested to support a business process that may have more to do with tradition than with best practices.

Therefore, it's best to focus on business outcomes over processes. "That's the way we've always done it" rarely cost-justifies developing and maintaining software modifications. After all, it's quite possible your software vendor and its existing customers may have thought of better ways to "skin the cat."

Sometimes change is necessary to fit a need. So, make sure you adequately specify the revision before programming is undertaken.

Surprisingly, users do not always know how they do something. When a specification is given to a programmer, what is specified is what will be produced.

Many times, however, users find out after they start using the modified application that there are some exceptions that were overlooked during the specification. The result, then, is more work and more cost.

Don't forget that these modifications must be maintained forever, which generally means spending the amount of money the modification costs about every five years or so to maintain it (unless your vendor incorporates enhancements into its standard supported product).

Data Migration & Conversion

Migrating data from your company's old system to the new one also presents an opportunity to "clean things up" and get organized. During this step, many companies improve important coding structures (such as G/L accounts and job cost coding) and clean up "clogged" vendor and/or customer master lists.

This is another area where focusing on business outcomes rather than on blanket statements (such as, "We want to bring over all job history" or, "We want to automate conversion of all our data") has been proven to produce better and more cost-effective results. Here are some questions to consider:

How Much Transaction History Do You Really Need?

Spelling this out can help you decide such important issues as how far back in time you want to go and what level of detail you want to bring over.

Our observation is that many companies spend a lot of time and money converting detailed history that is rarely, if ever, used. More and more companies find that the best practice is to bring over detailed history for active jobs and the current fiscal year, and summarize history for closed jobs and prior years.

Where Does Automated Conversion Really Make Sense?

The concept of having data flow automatically from the old system to the new one is intuitively attractive. The reality is that many data files (such as vendor, customer, and employee master information) are past due for clean-up. What better time to tackle this daunting task than when changing systems?

Does the Data Map Well from the Old System to the New One?

Ideally, your new system tracks additional, more detailed information than your old system. If a one-to-one field mapping does not exist between the two systems, some human intelligence will be necessary to manipulate the tables or to make appropriate changes after data conversion in the new system.

One way to aid in the conversion process is to extract data from the old system into a tool (such as Excel or a database management system), perform the necessary data cleansing, and add new fields if needed. Then, load the "scrubbed" data into the new system. Whichever method you choose, rest assured it will not be fully automatic.

How Will You Approach Reconciling & Verifying Your Data?

Ensuring key numbers (such as account balances, payroll history, etc.) are correct is an essential last step in the data conversion process. This must be done by the data owners – not the conversion programmer. Only the end users and stakeholders will know if their data has been converted correctly.

Who Will Perform the Conversion & Reconciliation Work?

Your personnel, vendor, or consultant are all options. Consider this wisely because conversion and reconciliation can be time-consuming and have less-than-desirable outcomes.

The "Go-Live"

Congratulations! You've planned, staffed, configured, trained, and converted – now it's time to flip the switch. You'll sleep better the night before if you're confident everything will work as planned on the big day.

Here are some steps to help make "go-live" day as smooth as possible:

- Develop a *pre-launch checklist* that details everything that has to happen, who will perform each task, and the sequence in which the tasks need to be performed for a successful go-live.
- Confirm readiness with a *mock go-live.* Conduct a go-live trial a couple of weeks before the target date. The new system will not be in place yet, so you can perform a mock implementation while the old system remains in use. This should help identify any kinks or gaps in your pre-launch checklist so you can adjust and be ready for the real switch.
- *Preplan your transition* from "implementation" to "support" mode with your vendor. Some vendors bring in independent contractors to do implementation work. Others have separate internal teams for implementation vs. ongoing support.

  During this transition stage, you may experience a change from a high level of vendor attention and familiarity with

your company to a steep learning curve as its support personnel takes over. Setting specific criteria, milestones, and procedures for this transition will smooth the way.

For example, "After we have run three live payrolls, a billing cycle, revenue recognition, and an accounting close in our product environment, we will be ready to transition from implementation consulting to ongoing support mode with our vendor."

In Conclusion

The central theme of this article: Rigorous exercise of routine project management disciplines is the recipe for a successful software implementation. Also, recognize that implementation is stressful and difficult, and that key people may experience "burnout."

So, here are some steps to ease these challenges inherent in just about every enterprise software implementation project:

- *Evaluate workloads constantly* to ensure that each participant is carrying his or her implementation workload.
- *Provide real support to the team members.* Recognize the team's efforts and sacrifices. This can be verbal mention in project newsletters or other company communication outlets, and/or rewards in the form of time off or bonuses after key milestones.
- *Don't let minor issues slow project momentum.* Henry Ford said, "Obstacles are those frightful things you see when you take your eyes off your goal."

  Spelling out your business objectives and outcomes – and keeping those in front of the team throughout the process – can help everyone drive through the inevitable issues.
- *Celebrate successes.* Throw a party or take the team out to a ball game – make it fun whenever possible.

The final point we hope you take away from this article: The time and money you invest to properly configure and implement your new enterprise software, and to train your staff to use it correctly, pays much higher dollar-for-dollar returns than the raw cost of the hardware and software. BP

LAURENCE C. TRUE is Managing Director at DR Construction Consultants, LLC in Clifton Park, NY, where he provides software implementation services to construction companies.

Larry has 33 years' experience providing estimating, project management and accounting services to the construction industry. He has a BS in Civil Engineering and is a frequent author and presenter for industry publications and associations.

A longtime CFMA member, Larry has served on the Board of Directors of CFMA's New York Capital Region Chapter since 2004 in capacities of Secretary, Program Committee, and President.

Phone: 518-369-7610
E-Mail: ltrue@drccllc.com
Website: www.drccllc.com

KURT M. KOENIG is Co-Founder and Vice President in charge of Product Strategy at Penta Technologies, Inc. in Brookfield, WI. He has more than 30 years' experience overseeing product development and applying computing technologies to the construction industry.

An alumnus of the University of Wisconsin, Kurt is a frequent writer and speaker on technology issues for the construction industry.

Kurt has served as an At-Large Director on CFMA's Board of Directors and as Co-Chair of the Technology Committee. He is currently a member of the IT Council, and the Conference Planning and Finance Committees. He also participates in CFMA's Leadership Mentoring Program.

Kurt is a member of CFMA's Milwaukee Chapter and of other industry associations, including AGC and ABC. A longtime, active member of AGC's Electronic Information Systems Committee, he also serves on the AGCxML Task Force and Review Committee.

Phone: 262-782-7700
E-Mail: kurt.koenig@penta.com
Website: www.penta.com