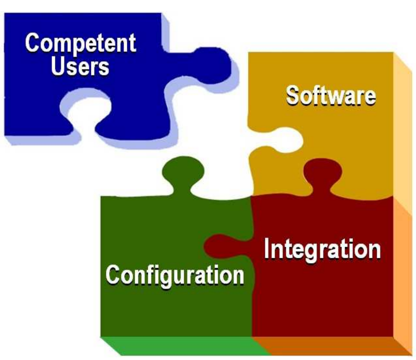
The Top 10 Pitfalls
of End User Training and How to Avoid Them

Dashe & Thomson
401 North 3rd Street
Suite 500
Minneapolis, MN 55401
www.dashe.com

Dashe & Thomson

The Top 10 Pitfalls of End User Training – and How to Avoid Them



Enterprise software projects can overcome all the technical challenges – but won't see a positive ROI until users become well-trained and competent.

Given the current state of the economy, businesses large and small are looking for ways to improve productivity while maintaining quality. For some, this means implementing or upgrading enterprise software systems – to streamline business processes, reduce costs, and improve productivity.

While it's one thing to turn on the software, it's quite another to be sure it's actually getting used by employees. That's where proper training becomes crucial. In fact, in order to reap the rewards of an enterprise system, businesses must be sure their end users are:

- Well-informed about the new system and its impact on their jobs
- Given the tools necessary to do their jobs in a new way when the system goes live
- Trained to use the new system effectively.

Unfortunately, businesses often underestimate the importance and complexity of the last point: effective training.

Once a business understands how critical training is for its enterprise software implementation, there are many decisions to be made about how, when, and where to train users.

Many of these decisions, if made hastily or without enough information, can backfire and create problems instead of solving them. On the other hand, if the training strategy is well-planned and executed, businesses can quickly realize the benefits of their enterprise software investment.

Pitfall 1: Designing an enterprise training program around a sketchy audience analysis

Training teams frequently build their training plans around an audience analysis that only gathers the number of users by 1) system role, and 2) security access level.

Obviously, this much information is not adequate to tell you anything but the number of classes you will need for each role. To design a quality training program you need to know much more about:

- The restrictions placed on users in their area of the business
- Each user's education and computer literacy levels
- How the implementation will change each user's job
- The practical aspects of each role.

The Solution:

One of the most effective tools for developing a training plan is the Design Impact Matrix, a tool that helps users understand which parts of their jobs will be most affected by the software and process changes, and by how much.

Business Unit Project Goal	Design Feature, Process Change or Major issue	Impact on Current Business Processes (H, M, L)	Job Roles Affected	Policy or Procedure change Required?	New Competencies Required to Perform Job	Training Needed to Affect the Change	Timeline Priority	Measures of Success
For example, to create a common Engineering Change Process	A software design feature, process change, or major issue that supports or hinders the goal	Degree of impact on current business processes	List of job roles affected and in what way	Any policy or procedure changes required to enforce new behaviors	List of new competencies / skills required based on the process or policy change	List of training experiences required to affect the change	When to implement the training	How do we define success and how will we measure it

The Design Impact Matrix tracks changes to job roles and processes, the degree of change, training required, policy changes, and more. Each change is tied back to a business objective to ensure training focuses on the priorities with the greatest business impact.

Dashe & Thomson

Taking the time to do a thorough audience analysis pays off in many ways. It is the best ammunition for justifying training choices. It allows you to design materials that meet all users' needs, from the practical needs of engineers in the lab, to the workers on the production line.

What's more, a lot of time and money can be saved if you don't have to rearrange the training schedule three times because no one thought to ask the accountants if the week before year end close is a good time for training.

Pitfall 2: Designing training first, setting standards later

There are several ways that companies build teams to develop enterprise software training:

- Bring in a consulting firm to do it all
- Bring in a consulting firm to supplement the in-house training team
- Engage independent training developers from a variety of sources
- Staff the project entirely with in-house resources.

In the first case, the consulting firm usually arrives with a methodology, a set of standards, and a development process. In the other three cases, the company must define and communicate the standards required. Unfortunately, these standards are frequently left undefined, resulting in a significant amount of re-work or re-creation of material.

Here's an example: A global food manufacturer recently rolled out an enterprise software program to a dozen facilities in three countries. The eight-member training team developed more than 1,000 procedures and process flows, which were used to create more than 50 different courses.

Stop and think for a minute what would happen if each of the eight team members had developed the training his or her "own way." The team would have found itself with a huge body of inconsistent materials – practically impossible to maintain long term – leading to a large post-project effort to re-work or re-create an entirely new training program.

Sometimes the worst case scenario is when the need for standards is discovered halfway through a project, and everything has to be reworked.

On another recent project, a major university engaged 10 enterprise software training developers, but gave no initial direction on standards. Five months into the project, after nearly 60 courses had been drafted, the client discovered the need for standards. This resulted in a full two months of rework to apply the standards to material that had already been written, adding more than $250,000 to the total cost of the project. For less than one-tenth of that cost, the organization could have assigned two people, for three weeks, to develop standards.

"Sometimes the worst case scenario is when the need for standards is discovered halfway through a project, and everything has to be reworked."

The Solution:

Take the time up front, always keeping your audience in mind, to set content, writing, and formatting standards that can be followed and enforced. This ensures the training and documentation output is consistent and easily maintained over the long haul.

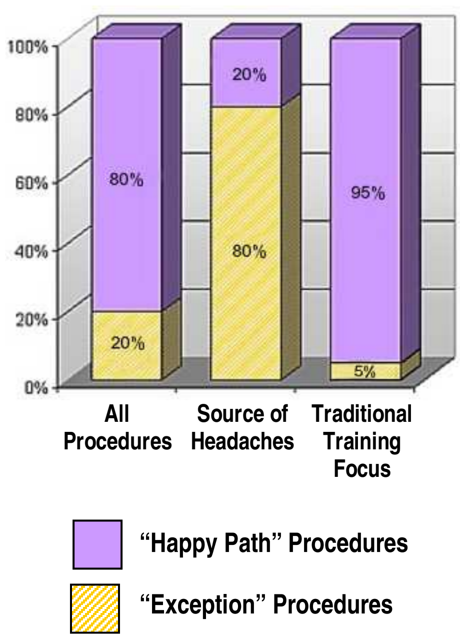
Examples of standards that should be defined include:

- Course elements and sequence, defining the use of learning objectives, demonstration, and exercises
- Look and feel, fonts, formatting, company branding elements
- Assessments and testing.

Pitfall 3: Staying on the Happy Path and ignoring exceptions

A big benefit of implementing enterprise software is the standardization it brings to business processes. This enables efficiencies and cost savings across parts of the business that did not interact before the new system.

Communication and training programs should certainly start by teaching users the *Happy Path* – that is, the new standard business process. However, anyone who has implemented enterprise software is well aware that one size never fits all   In fact, the 80/20 rule works well here; 80% of the transactions in your new system will follow the standard happy path and the other 20% will be exceptions.



The pitfall that many companies fall into is focusing only on the happy path and essentially ignoring the exceptions.

Unfortunately, it's the exceptions that can overwhelm help desks, leave users scratching their heads, and even bring operations to a grinding halt.

In addition, training that takes people down a single happy path, in which everything works perfectly, can bore learners, resulting in low retention rates and poor user acceptance.

The Solution:

Don't shy away from including at least common exceptions in your training program. This not only provides your users with the tools they need to follow the happy path, but also an arsenal to draw from when the happy path does not apply.

Give your users real-life scenarios where things get "messed up" and ask them to solve the real problems they'll have. The layer of complexity this can add to your training development pays off many times over with a pool of well-trained end users that are ready for anything, not just for following the happy path.

Pitfall 4: Lacking flexibility in your training design and delivery methods

This pitfall ties directly back to your audience analysis. A strong audience analysis helps you to easily recognize how different user groups may benefit from different types of training.

Difficulty arises when the project team ignores audience information, and designs a training program that offers only one delivery mechanism. Whether it's all instructor-led or all eLearning, over-dependence on a single training modality can dramatically hurt learning and retention.

For example, you may need to train 300 centrally located salespeople, 150 decentralized purchasing agents, and two people who need to enter and maintain master data records. If all three groups are trained in a classroom environment, the sales group would be pretty well served, the purchasing group would have the burden of huge travel costs and possible interruption of business in their absence. The two master data users, meanwhile, will either sit impatiently through class with one of the other groups, or will require a separate class just for them.

Blended learning programs combine training delivery methods to improve learner retention and expedite behavior change.

Whether it's all instructor-led or all eLearning, over-dependence on a single training method can dramatically hurt learning and retention.

The Solution:

Use the training approach or approaches that make the most sense for each audience and the content they need to learn. This is called blended learning.

Using the example above, instructor-led training is probably the right choice for the sales group. They do not need to travel for training and the classroom provides the opportunity for live interaction with the instructor to solve problems.

However, even though this group is centrally located, an eLearning module that covers the basics of system navigation would be ideal prework. This would save time for both instructors and students, and ensure that everyone is at about the same level of proficiency when they arrive for classroom training.

For the purchasing group, an online approach makes more sense, to minimize travel expenses and business disruption. The program for this group may consist of the same prework eLearning module used by the sales group, plus additional eLearning modules to train the core content, and a series of online webinars to answer user questions and demonstrate exceptions.

For the two master data users, the time and expense of creating a full blown classroom course or eLearning module does not make sense. These users may benefit most from being involved in system testing and working through content in a series on one-on-one meetings with subject matter experts.

And, of course, when training is complete and the system is ready to go live, an easily accessible reference system or library must be available to all users so they have the tools to help themselves in those first weeks the system is live.

Pitfall 5: Giving Subject Matter Experts (SMEs) too much influence on the training design

Training developers rely heavily on Subject Matter Experts – from both business and IT – to gather training content. These folks tend to feel a lot of ownership over the new system and have strong ideas about how to train users. This sense of commitment is generally a good thing. However, Subject Matter Experts are just that: *experts in their subject matter.* They are not necessarily experts in the process of training others.

By giving SMEs too much control over training design, you will likely end up with material that is long on exposition and lecture and short on practice. In addition, no two SMEs will bring the same approach, resulting in drastically different user experiences across functions. While allowing this to happen in the short term may seem easier and perhaps even effective, the training program becomes unsustainable as SMEs return to their regular positions or go to another project.

The Solution:

Train your SMEs early on how the training development process will work. Show them what the output will look like and why these design choices were made based on audience, time, budget, and sustainability. Get their buy-in before you begin gathering content in earnest. In addition, be sure that your training developers can articulate the rationale for the training design choices when meeting with SMEs.

When SMEs understand why training is being approached in a certain way and are on board with the choices, development can progress much more smoothly.

Pitfall 6: Designing a training program that lacks business context

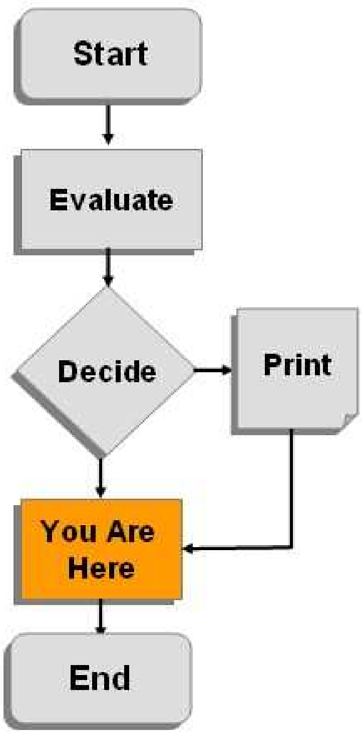
An enterprise system, by definition, integrates various aspects of your business into a single system and platform. Everything a user does in an integrated system can potentially impact another user downstream in the process.

For example, when one user creates a sales order, it creates demand, which in turn drives production planning, which drives raw material purchasing. Contrast this with a Word document created on a user's desktop: unless the user saves the document in an accessible network location, the Word document has no impact on the work of others in the company. This is why providing context, in terms of your own business processes, is so critically important. Users not only need to know how to use the new system, they must also understand when and, most importantly, why to use the new system.

Additionally, a user that does not have a clear understanding of their role in the overall business process is unlikely to help you realize the ROI for your enterprise system.

The Solution:
Provide the necessary context for your users with flow charts, business cases, and scenarios. Include clear section-introductions that explain the *when* and *why* for every procedure. Incorporate this information into your training and make sure you include it in your reference system or library.

We recommend creating an Electronic Performance Support System (EPSS), accessible directly from your enterprise software, to support your users at go-live and beyond. The EPSS is online help organized by business process. Users can view a high level business process and drill down through more detailed flowcharts or other contextual information to reach the procedure they need



Performance improves when employees understand how tasks fit within the overall business process.

to complete.

Whatever reference system you use, spend time teaching how to use the system. Empower users to help themselves by providing reference tools that provide context, not just keystrokes.

Pitfall 7: Underestimating the need for a well set up training environment

Overworked project teams, database administrators, and security teams will make the same argument every time: a dedicated training environment, they'll say, that mirrors current system configuration and contains valid sample data, is a waste of resources. Instead, they would prefer that an old sandbox or development client be used to avoid maintaining an up-to-date training environment.

Choosing to forego a real training environment is extremely short-sighted. When you learned to drive a car, think how it would have felt to take a driver's education course, spend a few hours in a simulator, pass your written test, wait six weeks, then get behind the wheel of a real car all by yourself. It would be scary, for you and everyone else on the road.

It's just as scary, from a project standpoint, to provide training in an environment that does not accurately reflect the production system. It's scary for your users and it's scary for your bottom line when your system goes live.

Invest the time, money and resources to create, populate, and maintain an adequate training environment, preferably with two to three instances available.

The Solution:

Invest the time, money and resources to create, populate, and maintain an adequate training environment, preferably with two to three instances available. **Multiple training instances allow the training team to maintain a "golden" instance that can be used to accept configuration changes and populate clean master and transactional data.** The golden instance is then used to refresh classroom and practice instances as often as necessary.

Create your training environment as soon as possible after the first wave of integration testing and updates have been completed. Provide access for project team members right away, and establish training and user IDs. These steps allow your training team to begin populating and testing the client with appropriate data.

SMEs can provide valuable assistance by furnishing a good set of data that the training team can then use to replicate the data for each training ID.

Finally, don't forget to budget enough time into your training plan for all of this to take place. The effort is less daunting than it might seem and the payoffs in user competency and confidence when your system goes live can be astounding.

Pitfall 8: Training your help desk too late or inadequately

Companies sometimes assume that it is sufficient to put help desk employees through regular end user classes to prepare them for their post-go-live duties.

The thinking goes that help desk users will just create tickets and route them to the project team until they have some time to pick up the finer points. This can easily backfire, leaving project team members overwhelmed by help tickets, a help desk that feels helpless, and users feeling frustrated.

Help desk personnel must be thoroughly trained prior to go-live to ensure a smooth transition.

The Solution:

Train your help desk early and often. If possible, involve them in testing to help build their confidence and competence with the system. Give them their own training curriculum that includes how to fix common problems such as locked user accounts, and advanced training on searching and using online help.

Challenge the skills of help desk personnel with realistic practice scenarios. The result will be a help desk that can solve a significant percentage of user problems without further escalation, leaving the project team free to focus on the more complicated issues. The help desk can also be your biggest supporter of a well-maintained EPSS or online help system. If the answer to a problem is readily available in online help, the help desk can direct the user to the information and reinforce the use of the system.

Pitfall 9: Underestimating training administration and logistics tasks

When companies are conducting a large training effort, especially across multiple locations or countries, there is a tendency to underestimate both the time and money it will take to:

- Schedule training
- Register users
- Manage translation efforts
- Launch eLearning courses
- Set up training facilities with appropriate hardware and software
- Coordinate printing and shipping.

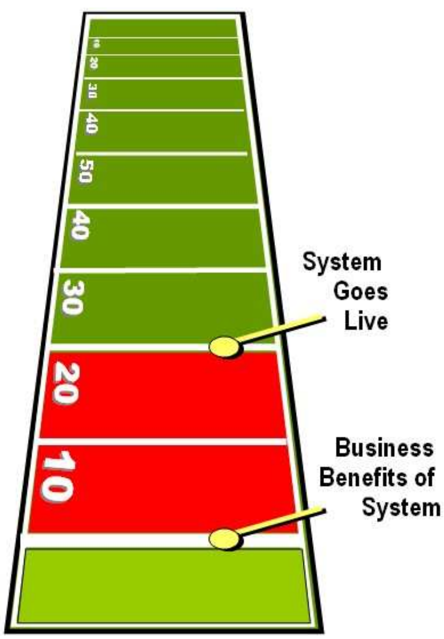
The Solution:
Dedicate at least one training team member to coordinate training administration and logistics, and work closely with training developers on development timelines, classroom system requirements, and course descriptions and timing. Depending on the extent of your training program, consider building in contingency funds for additional temporary staff to keep up with user registrations during the critical training period, or for outsourcing printing and shipping tasks

Pitfall 10: Treating the go-live date like it's the end of the project

As a project team works for months on a large-scale software project – putting in overtime, overcoming challenges, finding compromises – the go-live date takes on tremendous significance. This is natural in a sense – after all, everything revolves around the go-live date: workplans, resource availability, infrastructure decisions, and more.

After a certain amount of time, the project team – not intentionally of course – begins to treat the go-live date almost as if it represents the end of the project.

As anyone who has ever had to support users post-go-live can tell you: the go-live date is, in many ways, just the beginning. In fact, the go-live date can be compared to reaching the 20 yard line in a football game – you are entering the red zone. You don't realize any business benefits until you put the ball over the line, so to speak, which means getting users fully productive on the system.



The inability to think beyond go-live can result in a number of issues: First, implementation training materials often draw comparisons between the old and new systems to help users understand the changes. A new employee joining the company a month after go-live has no knowledge of the previous systems and processes, making your implementation training no longer effective.

In addition, failing to plan how reference materials will be maintained in the long term can be a costly error. Materials and eLearning courses used for implementation can quickly become out of date as new users and system enhancements enter the picture.

Consider your training program a two-phase project. Build materials and courses in phase one (that ends with go live) that can be easily leveraged in phase two (ongoing training for new users) with minimal rework.

The Solution:

As you design your training program, consider it a two phase project. Build materials and courses in phase one (that ends with go-live) that can be easily leveraged in phase two (ongoing training for new users) with minimal rework.

Next, gather additional audience information as it relates to new hires, such as turnover rates in certain positions. This information can help you determine if an ongoing course will need to be offered in a classroom once a quarter, or in a one-on-one situation once a week.

For your eLearning courses, be sure that you have either a maintenance contract in place or internal resources that can handle updates to keep the courses current.

Pitfall 11 (just for good measure): Not leveraging collaborative technology to build and execute your training plan

As businesses expand their reach, project teams and subject matter experts find themselves spanning the globe. Without real-time, collaborative tools, businesses increasingly find it difficult, if not impossible, to harness knowledge, align business practices, and realize a positive return on their software investments.

Here's an example: a five-person training team needed to collaborate with and gather training content from subject matter experts based in Argentina, England, Australia, and India. Time zone issues aside, gathering SME content through the "normal" modalities of meetings, conference calls, and email was daunting. A different solution was needed.

On another project, the training team was unexpectedly given the task of creating a knowledge base for the soon-to-be Help Desk – four weeks prior to go-live. No small task considering the Help Desk was to support over 7000 employees as they converted big-bang style to six new tier-one applications.

The Solution:

For both of these projects, leveraging Web-based collaborative technology was the answer.

In the first example, a SharePoint site was set up with specific metadata to help manage document flow and assignments between the project team and the training team. The SharePoint site included a project calendar showing review, printing, and shipping deadlines by country, a collaborative workspace for editing documents and posting messages to the entire team, and specific reporting views to avoid the need to track documents outside of the site.

In the second example, the training team inventoried the content they had developed over the course of the project, and, within 48 hours, repurposed the content for the Help Desk and populated a wiki with over 300 searchable articles. The Help Desk personnel were trained on how to use the wiki, and the Help Desk was successfully up and running at go-live. During the first months after go-live, the front page of the wiki served as a daily hot-list, and new searchable articles were added on-the-fly by Help Desk personnel. Perhaps the real power of the knowledge base came after the Help Desk was disbanded, and the wiki was repurposed for end-users, laying the groundwork for an ongoing, single-source knowledge repository.

While collaborative technologies are still relatively we must embrace them to achieve the returns we expect from enterprise software investments.

In Conclusion:

For most companies, training end users on an enterprise software application is the largest single training effort they will ever attempt. It's the training Olympics so to speak: you can prepare for years, but when the big day comes, you only have one shot at getting it right.

Making yourself aware of the common pitfalls associated with enterprise software training is the first step. If you plan carefully and take steps to avoid them, your chances for success increase dramatically.

For More Information Contact:

Paul Paradis
paul.paradis@dashe.com

Dashe & Thomson
401 North 3rd Street
Suite 500
Minneapolis, MN 55401

612-338-4911
www.dashe.com