4

Knowing Isn't Doing

THE REASONS WHY MOST E-LEARNING (AND OTHER TRAINING, FOR THAT MATTER) IS SO BAD— AND FIVE QUESTIONS TO ASK TO BEGIN TO MAKE IT BETTER.

Everyone seems to want to do e-learning these days. They have all been told how to buy the right stuff to get started, and perhaps they are having fun trying to do it. Occasionally companies find they are not having fun, and when that happens sometimes they call me. People don't usually admit to being in over their heads, so it was rather strange to get a call about a company that had been building e-learning for a while but thought they were doing a bad job. I mean, who ever thinks they are doing a bad job at anything? Maybe that is just an American attitude. This was a German company.

This was a large utility company. They had decided that e-learning was important to their future, so they did what most big companies do, they told one of their HR guys to investigate and gave him a budget. He attended conferences, he heard sales

pitches, he hired people, and soon enough he was building some e-learning programs.

After a while he began to feel that he had been had. He needed a learning management system. He needed the services of a variety of companies who wanted to build e-learning for him. He acquired various tools, books, consultants, and employees, but in the end he felt that what he was being sold wasn't very good and therefore what he himself was building wasn't very good. He invited me to see what he and his team had built.

I warned him that I was going to hate it, as I warn everyone whose software I am about to review. I always hate the software I see. "Why," you wonder, "haven't you ever seen anything you liked?" Well, in a word . . . "No."

Of course I am a harsh judge, but the reason that I am always unhappy with what I see is more complex than that. No matter how many of them I see . . .

Today's e-Learning or Educational Software Programs Always Seem to Contain Major Design Flaws

Flaw #1. Telling, Not Doing

One flaw I see every time is that telling tends to dominate the experience. Everyone seems to know that people learn by doing, but somehow e-learning designers can't seem to remember it. Maybe it's because people keep telling it to them. Every e-learning program seems to want to tell you the company policy or have the president of the company welcome you to the system, or tell you the right thing to do and then ask you to do

it, or ask you a question and then tell you the right answer. Or, as was the case with my German client, they gave a lot of their budget over to a graphic arts company that created the coolest little animated character who showed you around an otherwise mind-numbing system.

Flaw #2. Cleverness, Not Fun

e-Learning designers know that e-learning should be fun, but the material they are teaching is usually not a laugh riot. So they introduce a cartoon character, spend lots of money on first-class animation, and have the character introduce topics in what is supposed to be a fun and satisfying way. And while these characters are quite often rather appealing, their educational value hardly justifies the money spent on them. In a sense the software is just a come-on: It seems appealing, it looks like it is going to be fun, but all the animation and humor is intended to hide the dull experience you are about to have with actual material to be learned.

Flaw #3. No Stories

A third common flaw is the lack of good storytelling. This one is odd because it is so easy to avoid. The first "telling, not doing" flaw is actually hard to figure out how to get around. The "animated character" flaw is a more subtle trap because on the surface there seems to be nothing wrong with incorporating humor and clever graphics into your learning software. The "storyless" flaw is harder to understand. How hard is it to collect a few good stories from the company's collective experience and put them in the software? No matter how boring the soft-

ware, at least people would remember a well-told story, especially if it were told by someone they knew or it related to some experience they had had. But invariably the e-learning designers, desperate to put learning objects into their learning management system, have left out the stories, which was all anyone was going to remember anyway.

Flaw #4. Relentless Quizzes

The fourth flaw is the most common. e-Learning designers love quizzes it seems. Every other screen seems to be a test of some sort, asking you what you want to do or asking you what you think the right answer is to some issue you don't care about or based on something you supposedly just read or in something you are about to do. School is all tests these days, so I guess e-learning designers think that is because educational theorists think quizzes are the essence of good instructional design, when the real reason has to do with every politician's current obsession with measurable outcomes.

Meanwhile, Back in Germany, They Want Help . . .

To make a long story short, this German utility company showed me software that had a cute animated character telling you things you didn't want to know and asking you questions you didn't care about the answer to, about a system you didn't want to learn how to use all that much in the first place, and weren't going to learn to use by being told about it . . . all with nary a story to be found.

The only unusual part of this story is that the man who was in charge of it all didn't much like his programs either. He wanted help.

HOW DID THE GERMAN HR GUY KNOW THAT HE NEEDED HELP? *His customers (the people in the courses he built) were complaining, but that's not as bad as it might sound. Quite often, such complaints are explained away by stating that the customers were not ready for e-learning or saying that they would have to get used to it because the company decided it was more economical to do it this way. This is the point where some compromise is usually made and everyone agrees that a "blended approach" would be best—which means as far as I can tell that they won't get rid of all their classes and thus will save a few training jobs.*

I think he knew that things were bad because he had read my book (Designing World Class e-Learning) *and realized that he had made every mistake I outlined. Although I might note that plenty of others have read this book under similar circumstances and not drawn the same conclusion.*

So I arrived in Germany with the task of teaching his team . . .

How to Do e-Learning Right in Three Days

I started by talking to them about learning, which I did because they expected it. (As you may know by now, I really don't think that telling people about learning will cause them to change their behaviors one iota.) I showed them some educational software that I think is good and we discussed what might be improved and how it compared to what they had built. And while discussions like this are good, still I knew that this wasn't going to help them learn how to do it any better either.

What next?

Start with a Few Rules of Thumb

My favorite rules of thumb for building e-learning are to *ask experts about what goes wrong in their companies* and to *start people thinking about training as a kind of just-in-time remediation,* instead of school that prepares you for events that will take place years later after you have long forgotten the preparation.

> RULES OF THUMB FOR BUILDING E-LEARNING
>
> - Ask experts about what goes wrong in their companies.
> - Start people thinking about training as a kind of just-in-time remediation.
> - See e-learning as being about doing.

But mostly I try to get them to *see e-learning as being about doing.* It is about doing in two ways, of course. One issue is what employees will learn to do better. Another is what designers must learn to do differently. The best way to approach both of these at the same time is to create a simulated design exercise. *Instead of talking about what they were doing wrong, I suggested that we simply start to do something right.*

I asked them what their next e-learning project was going to be and they said new hire training, so I suggested we start the process together the next morning. I asked them to bring the expert on new hires to the table.

Bring on the SME

The team I was training sat across from the expert, and I sat to the side. I said that I would comment on what they were doing as I saw fit.

The fact is they had no idea what to do at this point. They had never started a project in this way. I mentioned that getting stories from the expert was an important first step to building any training program and understanding what needed to be taught was another and that the two were quite related. I suggested they start.

They sat there frozen.

I said, "Well, ask him a question."

"We don't know what to ask" was the somewhat testy response.

"Well, what would you need to find out before you start?" I said.

There was a brief discussion, and finally one of the more aggressive members of the team asked: "What does a new hire need to know?"

> HADN'T THEY BEEN LISTENING? *I almost fell over. Here I had been talking about learning by doing for a full day and it really hadn't penetrated. I had talked about the difference between recognition memory and recall memory. I had talked about the difference between conscious knowledge and implicit knowledge. I had talked about "knowing how" versus "knowing that."*
>
> *None of this had penetrated.*
>
> *Why I found this surprising I don't know. I always say that you can't learn by listening. Somehow I supposed that didn't include listening to me.*

The expert responded that there was a new hire manual and that everything that a new hire needed to know was contained in it. Next question.

I loved that guy. What a great answer. He also knew that it was a stupid question.

I asked the team what was wrong with the question they had asked. They responded defensively, saying that they needed to know what new hires needed to know and so they asked and what was wrong with that?

I said: "Well, it is an awful question. Can you figure out why?"

They had figured out that it was bad question from the expert's response, but they really didn't know how to improve on it.

I suggested they . . .

Try Thinking About Doing Rather Than Knowing

They asked, "What should a new hire know how to do?"
The expert responded with an absurdly long list of things, all of which were in the new hire manual as well.

I asked if they could figure out what to do.

They could not.

We sat there in silence.

Finally I said: "What is the biggest mistake that new hires make when they are first on the job?"

The expert's face lit up. He began to tell a story about some complex software that kept breaking and how new hires were clueless about how this software worked and so they couldn't answer customers' questions when they called in to complain about the software.

Huh?

I thought this was a new hire program. Rules, benefits, how we hate sexual harassment, that sort of thing.

"Nah," he said (or whatever the German equivalent was).

"What is it that these new hires are doing?" I asked.

"They answer the phone," he told me.

"Really? What about?"

"We sell software to other utility companies. It is complex and doesn't always work. Companies call up and ask difficult questions. The new hires can't know the answers right away, so they don't do so well for a long while, until they get the complex software issues down."

"So you don't want a new hire training program at all," I said.

"We don't?"

"No, you need a program to train people to answer the phone and do customer service. We have done plenty of those kinds of programs. They have nothing in common with new hire training."

And there you have it. *The reason why most training is so bad.* Unbelievably often, the people who write the training do what they were asked to do. They build what they were asked to build. But the people who are doing the asking don't always know what they want.

> *The reason why most training is so bad. The people who write the training do what they were asked to do. They build what they were asked to build.*
>
> WHAT TO DO INSTEAD
>
> *Think about what the company really needs, not what they ask for.*

Moreover, the designers of the training don't know how to ask the right questions to find out what the company really needs. *Learning to interview experts is the cornerstone of the process of the design of training.* Teaching your designers to interview is more important than teaching them to build software.

What Are the Right Questions to Find Out What Is Really Needed?

1. What are employees having trouble doing properly?

I am not sure that the others matter all that much given the overall significance of number 1. But here are some more:

2. Can you tell me a story of when an employee didn't know what to do and it caused a big problem for the company?
3. Under what circumstance do employees do the wrong thing, even though they have been told how to do the right thing?
4. What problems are causing the company real trouble right now?
5. What are the key things an employee needs to know how to do in this company?

These five questions have worked well for us over the years. They are reasonable things to ask experts, and they usually produce responses you can work with. The questions about problems are especially important for designing training. Every time you find a problem you have an opportunity for training.

Well, almost every time . . .

Some Things Simply Can't Be Taught

At Wal-Mart they had all kinds of fun problems to tell us about when we asked our questions. My favorite involved a cashier interrupted by the phone while she was checking a customer out. She kept talking on the phone and the customer started getting angry. This didn't stop the phone conversation, however. The

cashier just kept on talking. So the customer's complaints just kept growing louder. The cashier's solution was to hit the customer over the head with the phone. A novel method of customer relations to be sure.

So now that we know this error, what do we do about it? Do we create a module where you are told never to hit the customer, no matter how angry you are? Or do we discuss how destruction of company property is a bad thing? Do we design a course in anger management? Do we simply ignore this incident as a one-time oddity?

It may very well not be an oddity, but we do have to ignore it. We cannot train people to avoid playing the fool. Some folks are simply foolish.

So the questions were good ones to ask, and the answer was great fun to hear, but it didn't result in a teaching point. Some things simply can't be taught.

◊ ◊ ◊

Jump Start Your Training

Some do's and don'ts:

- Don't tell anybody anything in an e-learning program.
- Don't use graphics just because they are cute.
- Don't hide boring material in a pretty costume.
- Don't forget that the software itself must tell a story.
- Don't use e-learning as an excuse to create a multiple-choice test.
- Don't confuse yourself with the idea that "blended learning" is any more than a political compromise.

- Do remember to ask experts about what isn't working in the company.
- Do ask what employees need to learn to do better at their jobs.
- Do ask what instructional designers are doing wrong.
- Do think of e-learning as "just-in-time remediation."
- Do ask experts for their "horror stories."
- Do not imagine that all horror stories teach an important lesson.
- Don't ask what an employee needs to know.
- Do ask about what employees will be expected to be able to do.
- Don't confuse employee status with employee function—doing errors transcend status.
- Don't think that those who have requested training know what they actually need.
- Do start questioning experts about problems in the company's business.
- Do learn to interview experts properly.
- Do focus training designs on teaching points.